

# **Archivierung und Reorganisation buchungsrelevanter Schnittstellendaten aus SAP R/3**

Prüfer: Prof. Dr. Faeskorn-Woyke

## *Themen*

## **Fachhochschule Köln Abt. Gummersbach**

**Fachbereich Informatik - Studienrichtung Wirtschaftsinformatik**

## **Diplomarbeit**

**von**

**Stephan Protogerakis**

**Matrikelnummer: 10233387 1 6**

# **Archivierung und Reorganisation buchungsrelevanter Schnittstellendaten aus SAP R/3**

**Referent: Prof. Dr. H. Faeskorn-Woyke**

**Koreferent: Prof. Dr. F. Victor**

**angefertigt im Sommersemester 1999  
eingereicht am 01.07.1999**

1. Vorwort
2. Einleitung
3. BAYER
4. SAP
5. Input-Verfahren
6. Probleme bei der  
Datenübernahme
7. Archivierung
8. Reportlogik
9. Entwicklungsumgebung
10. Der Abstimmreport
11. Hauptprobleme
12. Schluß
13. Anhang

# VORWORT

In dieser Diplomarbeit wird die Lösung eines Standardproblems aus dem Umfeld des Rechnungswesenmoduls des SAP R/3-Systems behandelt.

Die der Diplomarbeit zugrundeliegende Aufgabenstellung gliedert sich wie folgt: Buchungen aus Vorsystemen werden in das R/3-System (fast) automatisch, nämlich per Batch-Input, eingespielt. Die dabei gebuchten Daten können von den angelieferten abweichen. Um der internen Revision diese mögliche Abweichung ausweisen zu können, müssen bisher die Protokolle der BI-Mappen aufgehoben werden.

Die Protokolle stehen in einer großen sequentiellen Datei, der sogenannten BI-Protokolldatei, die nicht auswertbar oder archivierbar ist. Daher muß eine andere Möglichkeit zum Nachweisen dieser Differenzen entwickelt werden. Dieser Nachweis muß dann archiviert werden können, damit die BI-Mappen und die dazugehörigen Protokolle gelöscht werden dürfen. Ziel ist also:

1. eine Reportlösung zu entwickeln, die Abweichungen der im SAP-System verbuchten Daten zu den Daten aus Vorsystemen revisionssicher nachweisen kann.
2. ein Konzept zu entwickeln, nach dem die Reportliste in einem nächsten Schritt archiviert werden kann
3. die BI-Mappen mit den dazugehörigen Protokollen zu löschen
4. die BI-Protokollsatzdatei zu Reorganisieren

Die Aufgabenstellung dieser Diplomarbeit ist eine von diversen Aufgaben, die im Zentralen Servicebereich Informatik der BAYER AG in Leverkusen als aktuell eingestuft wurden. Zu Beginn meiner Arbeit lagen einige theoretische Ansätze vor, jedoch kein ausgereiftes Konzept und kein Coding in ABAP/4.

Ziel war es, einen möglichen Arbeitsablauf, der im täglichen Rhythmus vonstatten geht und die Protokolldatei von SAP R/3 auf ein erträgliches Maß an Größe schrumpfen läßt, vorzubereiten und einzuführen.

Der Report, auf den im einzelnen noch eingegangen wird, wurde in der SAP-Version 3.0D in der Entwicklungsumgebung ABAP/4 erstellt. Während dieser Zeit wurde parallel der Releasewechsel auf 4.0B realisiert.

Nach einigen Tests wurde sichergestellt, das der Report auch im Release 4.0 vom Funktionsumfang her fehlerfrei ablauffähig ist.

Die Migration hat einige kleinere Änderungen in Details der Logik mit sich geführt, die jedoch den reibungslosen Reportablauf nicht in Frage stellen.

In dieser Arbeit werden einige Fremdwörter verwendet, die sich zum Teil im Anhang unter dem Oberbegriff Glossar wiederfinden. Die erklärten Begriffe sind

jeweils beim ersten Auftreten im Text mit einem Stern (\*) gekennzeichnet.

Alle in dieser Arbeit genannten Produkte wie SAP, Oracle u.ä. sind geschützte Produkte und unterliegen dem Urheberrecht.

Die meisten der verwendeten Abbildungen sind Screenshots aus dem SAP R/3-System der BAYER AG. Bei diesen Abbildungen wird auf die Quellenangabe verzichtet.

# EINLEITUNG

Unter dem Schlagwort Business Process Reengineering wird seit geraumer Zeit im Unternehmen eine Anpassung oder Neuorganisation der Geschäftsprozesse\* vorgenommen.

Eng verbunden mit diesem Schlagwort ist die betriebswirtschaftliche Standardsoftware SAP R/3 der Walldorfer Firma SAP AG.

Die Implementation dieser Software ist bei größeren Unternehmen nicht trivial. Viele SAP-zertifizierte Berater sind häufig dauerhaft vor Ort und helfen bei der Einführung. Viele Begriffe des heutigen Informationsmanagements wie Vorgehensmodell, Workflow\* und Einführungsleitfaden finden sich hier wieder, deren Ursprung das Problem der Einführung von ERP\*-Standardsoftware war. Es ist ein komplexes und sprachlich schwieriges Themengebiet und es vergeht einige Zeit, bis die SAP-Systeme stabil und flächendeckend eingesetzt werden können.

Bei BAYER existieren eine Reihe von Legacy\*-Systeme, deren Daten im Vorsystem MBS\* gesammelt werden, und dann per Batch-Input-Verfahren in SAP eingespielt werden.

Die Arbeit gliedert sich in folgende Bereiche:

In Kapitel 3 wird der Bayer-Konzern in seiner Unternehmensstruktur mit einigen Daten erläutert.

In Kapitel 4 wird das Unternehmen SAP vorgestellt. Desweiteren wird das Produkt SAP R/3 mit seiner Geschichte und seinem Aufbau angerissen. Dabei wird dann ein Baustein herausgenommen und exemplarisch erläutert. Wegen des kausalen Zusammenhangs zu dieser Diplomarbeit wurde hierzu der Baustein FI\* ausgewählt.

In Kapitel 5 wird auf die Input-Verfahren eingegangen, die von SAP zur Einstellung von Fremddaten angeboten werden. Auf das Batch-Input (BI) Verfahren wird genau eingegangen, da es die wesentliche Schnittstelle der Arbeit darstellt. Um das der Arbeit zugrundeliegende Problem zu verstehen, sind einige Grundkenntnisse dieses Verfahrens notwendig. Als kleiner Exkurs wird erläutert, was bei einem Buchungsvorgang SAP-intern geschieht.

In Kapitel 6 wird beschrieben, an welchen Stellen im Organisationsablauf MBS-Daten verändert werden können. Die Funktionsweise dieser Stellen wird ebenso beschrieben wie die Problematik, diese Fälle nachzuweisen.

In Kapitel 7 wird das Thema der Archivierung allgemein und speziell bei FI-Belegen aufgegriffen und kurz erläutert.

In Kapitel 8 wird das eigentliche Lösungskonzept des Reports zur Abstimmung der MBS mit den SAP-Daten beschrieben: die inhaltliche Logik der Problemstellung, die Lösung, und in welchen Schritten sich die Lösung vollziehen soll.

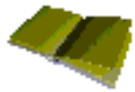
In Kapitel 9 wird die integrierte Entwicklungsumgebung des SAP R/3-Systems beleuchtet.

Ein kleiner Exkurs widmet sich dem Korrektur- und Transportwesen des SAP R/3-Systems.

In Kapitel 10 wird der inhaltliche Aufbau des eigentlichen Reports erläutert, Programmsegmente werden erklärt und das Zusammenspiel der Fremddaten mit den SAP-Daten aufgezeigt.

In Kapitel 11 wird auf die Hauptprobleme eingegangen, die bei der Erstellung dieser Arbeit auftraten.

Kapitel 12 faßt die Arbeit zusammen und gibt einen kleinen Ausblick auf die Zukunft.



# BAYER

Die vorliegende Arbeit wurde vollständig bei der BAYER AG im Werk Leverkusen erstellt.

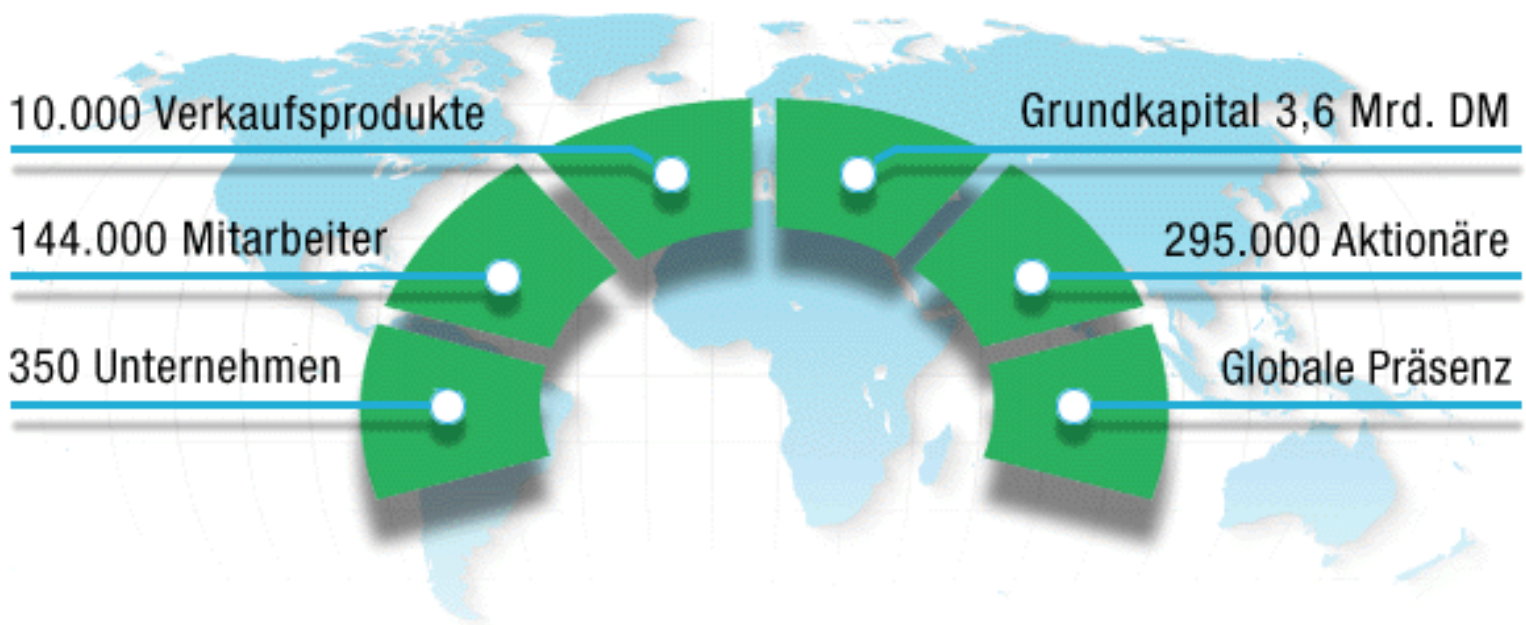
Der Bayer-Konzern ist ein diversifiziertes, internationales Unternehmen der chemisch-pharmazeutischen Industrie. Es bietet ein breites Sortiment von Produkten und Leistungen, das von den Bereichen Gesundheit und Ernährung über Kunststoffe bis zu Spezialprodukten in der Chemie und Informationstechnik reicht. 350 Gesellschaften auf sechs Kontinenten sind dem Konzern angehörig (vgl. Abb. 1).

Die Aktivitäten der BAYER AG sind in fünf Arbeitsgebieten zusammengefaßt: Gesundheit, Landwirtschaft, Polymere, Chemie und Agfa.

Insgesamt 21 Geschäftsbereiche sind als Profitcenter für ihr operatives Geschäft verantwortlich. Daneben gibt die Konzernbereiche bestehend aus 7 Bereichen und die Zentralen Service Bereiche in 7 Bereiche untergliedert.

Die Zentrale des Bayer-Konzerns ist in Leverkusen. Sie ist zugleich Sitz von Vertrieb und Verwaltung der AG. Das Leverkusener Werk erstreckt sich über ein Gelände von 3,4 Quadratkilometern mit rund 600 Gebäuden und mehr als 30.000 Mitarbeitern (Quelle: BAYER-Intranet).

## Bayer im Überblick



**Abbildung 1 - Allgemeine Bayer-Konzern-Übersicht [5]**

Gemäß Vorstandsbeschuß soll SAP konzernweit als Strategische Software zur Abdeckung

betriebswirtschaftlicher Vorgänge eingeführt werden.

Es gibt eine Vielzahl von betriebswirtschaftlichen DV-Systemen, deren Daten in einem Buchungssystem zusammenlaufen.

Nicht alle DV-Systeme laufen in das SAP-Vorsystem hinein.

Die Kreditorenbuchhaltung\* ist im SAP-System der BAYER AG gut implementiert, deshalb laufen Daten aus anderen DV-Systemen, die die Kreditoren betreffen, in das SAP-Vorsystem ein.

Bei der BAYER AG sind verschiedene R/3-Systeme im Einsatz, auf denen ein oder mehrere Arbeitsfelder abgebildet werden (s. Abb. 2). Daten können so zwischen den Arbeitsfeldern ausgetauscht werden, sofern diese in einem fachlichen Kontext zueinander stehen.

Den Arbeitsfeldern zugrunde liegt ein prozeßorientiertes (logisches) Verteilungskonzept, in dem die Arbeitsfelder die einzelnen Segmente bilden. Arbeitsfelder umfassen i.d.R. ganze Geschäftsprozeßketten und bilden weitestgehend kapselbare Einheiten für ein auf die Optimierung von Geschäftsprozessen ausgerichtetes Reengineering.

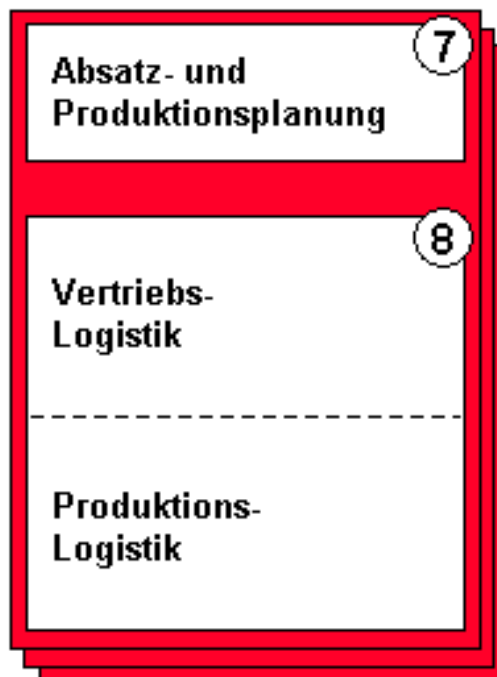
Datenkommunikation zwischen den Arbeitsfeldern geschieht entweder über die SAP-SAP\*-Schnittstelle (ALE\*), oder über Vorsysteme (z.B. MBS). Sowohl SAP-Buchungen als auch Buchungen aus anderen DV-Systemen müssen dann per Input-Verfahren im Zielsystem "gebucht" werden, damit Einträge in den entsprechenden Tabellen generiert werden. Hierbei treten jedoch diverse Probleme auf.

Mittlerweile gibt es 12 verschiedene Arbeitsfelder, die sich wie folgt gliedern:

#### Global Konzern



#### Regional/Lokal Geschäftsbereich



#### Lokal Gesellschaft

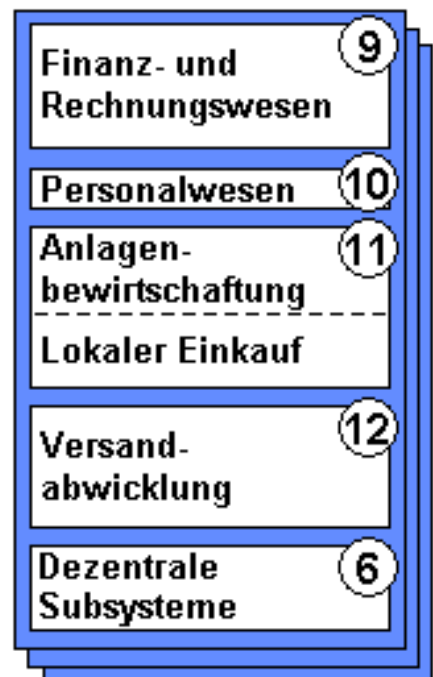


Abbildung 2 - Arbeitsfelder der BAYER AG [5]

Die Abteilung IM-KSB betreut die Arbeitsfelder 9 und 11. In der Diplomarbeit ist die Rede von

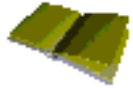
einem Abstimmreport; dieser bezieht sich auf das Abstimmen von Buchungen aus dem Legacy-System MBS auf dem Host und den Buchungen im Arbeitsfeld 9.

Die sogenannten SAP-SAP-Buchungen beziehen sich auf die Schnittstelle zwischen den Arbeitsfeldern 11 und 9 (Daten aus AF 11 gehen in AF 9).

Die R/3-Systeme für die restlichen Arbeitsfelder werden sukzessiv eingeführt.

Ein weiteres Augenmerk liegt auf dem Arbeitsfeld 8 mit den Modul SD, MM und PP-PI.

Dieses Arbeitsfeld ist mit großem Abstand das mächtigste. Es ist nur mit realisierbar, wenn alle anderen beteiligten Module wie z.B. FI bereits aktiv sind. Auch bezüglich der Systemleistung stellt dieses Arbeitsfeld andere Dimensionen dar. So ist beispielsweise ein Datensatz in SD über fünffach so groß wie ein Datensatz in FI.





# SAP

## Das Softwareprodukt SAP

### Produktentstehung

### Produktbeschreibung

## Der Baustein FI (Finanzwesen)

### Belegprinzip und Belegsteuerung

### Organisationsstrukturen im Modul FI

### Integration mit anderen SAP-Anwendungen

### Bearbeiten von Belegen im Modul FI

### Reporting

---

Die SAP AG als Muttergesellschaft der SAP-Gruppe ist das weltweit führende und fünftgrößte Softwareunternehmen mit über 30 Prozent Marktanteil und mehr als 8.000 Kunden in über 85 Ländern.

Das Softwarehaus wurde 1972 mit Sitz in Walldorf / Baden gegründet.

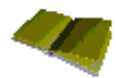
Seit 1979 gibt es R/2, das gemäß der damaligen Philosophie Mainframe-orientiert war (MVS, BS2000).

Seit Sommer 1992 ist R/3 auf dem Markt. Das Produkt war ursprünglich für mittelständige Unternehmen konzipiert, die sich im Zuge des Downsizing Client-Server orientieren.

Mittlerweile wurden mehr als 12.000 R/3- und R/2-Installationen weltweit durchgeführt.

Das Unternehmen hat über 11.000 Mitarbeiter und einen Umsatz von ca. 4 Mrd. DM.

(Quelle: SAPNET)

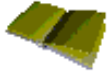


## **4.1 Das Softwareprodukt SAP**

### **4.1.1 Produktentstehung**

Der Druck durch die Globalisierung der Märkte, der zunehmende internationale Wettbewerb, kürzere Produktlebenszyklen, höhere Anforderungen an Qualität und Flexibilität der Produktion bzw. Produkte und geringere Gewinnmargen zwingen Unternehmen vermehrt zu strategisch wichtigen Entscheidungen. Aufgrund der Komplexität der Probleme und der Dynamik des unternehmerischen Umfelds ist der Bedarf an Informationen als Grundlage für die Entscheidungsfindung gestiegen. Die Qualität der Informationen spielt hierbei eine wesentliche Rolle und bestimmt oft über Erfolg oder Mißerfolg einer Unternehmung. Viele Unternehmen verfügen nicht über eine integrierte Software, so daß Informationen oft nicht konsistent an vielen Stellen des Unternehmens auftauchen. Es sind nicht wenige Firmen, die diesen Informationsmißstand noch nicht bemerkt haben. Die Informationsverarbeitung ist zu einer strategischen Wettbewerbskomponente geworden. Eine qualitativ hochwertige Software kann die nötigen Informationen für die unternehmerischen Entscheidungen zur Verfügung stellen und wirkt sich mehr denn je auf den Erfolg am Markt aus.

Das Softwarehaus SAP AG versucht, den oben erwähnten Informationsmißstand vieler Unternehmen mit dem Standardsoftwareprodukt SAP R/3 entgegenzutreten, in dem neben isolierten Funktionalitäten der einzelnen betriebswirtschaftlichen Komponenten die Integration und die Abhängigkeit einzelner betriebswirtschaftlicher Vorgänge im Vordergrund stehen.



#### 4.1.2 Produktbeschreibung

Die SAP-Gruppe entwickelt integrierte Softwarepakete, die die Anforderungen der Unternehmen erfüllen, nämlich betriebswirtschaftliche Funktionen, deren Lösung bisher meist durch unternehmensspezifische Individualprogramme erfolgte, durch Standardsoftware abzubilden. Die Probleme dieser Individualsoftware in Bezug auf Entwicklungskosten, Weiterentwicklung, Datenredundanz\* und –inkonsistenz\* sollen durch eine einheitliche Systemumgebung mit modularem Aufbau beseitigt werden.

Grundgedanke der SAP-Software ist die funktionale Abdeckung aller wesentlichen Unternehmensanforderungen sowie die umfassende Integration betriebswirtschaftlicher Datenverarbeitung. Die Aufteilung der Software in einzelne betriebswirtschaftliche Bausteine ermöglicht einen flexiblen Einsatz der Komponenten, abhängig von den unternehmensindividuellen Anforderungen. Als betriebswirtschaftliche Software bildet R/3 in über 800 Geschäftsprozessen die Abläufe innerhalb eines Unternehmens durch teilweise auch eigenständig einsetzbare Module ab, deren Gesamtnutzen aber erst durch die Integration mehrerer Module richtig zur Geltung kommt. Einige Module, die in Unternehmen zusammenspielen, sind:

- Finanzwesen (FI)
- Anlagenbuchhaltung (FI-AA)
- Controlling (CO)
- Projektplanung (PS)
- Materialwirtschaft (MM)
- Instandhaltung (PM)
- Vertrieb (SD)
- Qualitätssicherung (QM)
- Produktionsplanung und –steuerung (PP)
- Personalwesen (HR)
- Office-Funktionen (OC)
- Investitionsmanagement (IM)

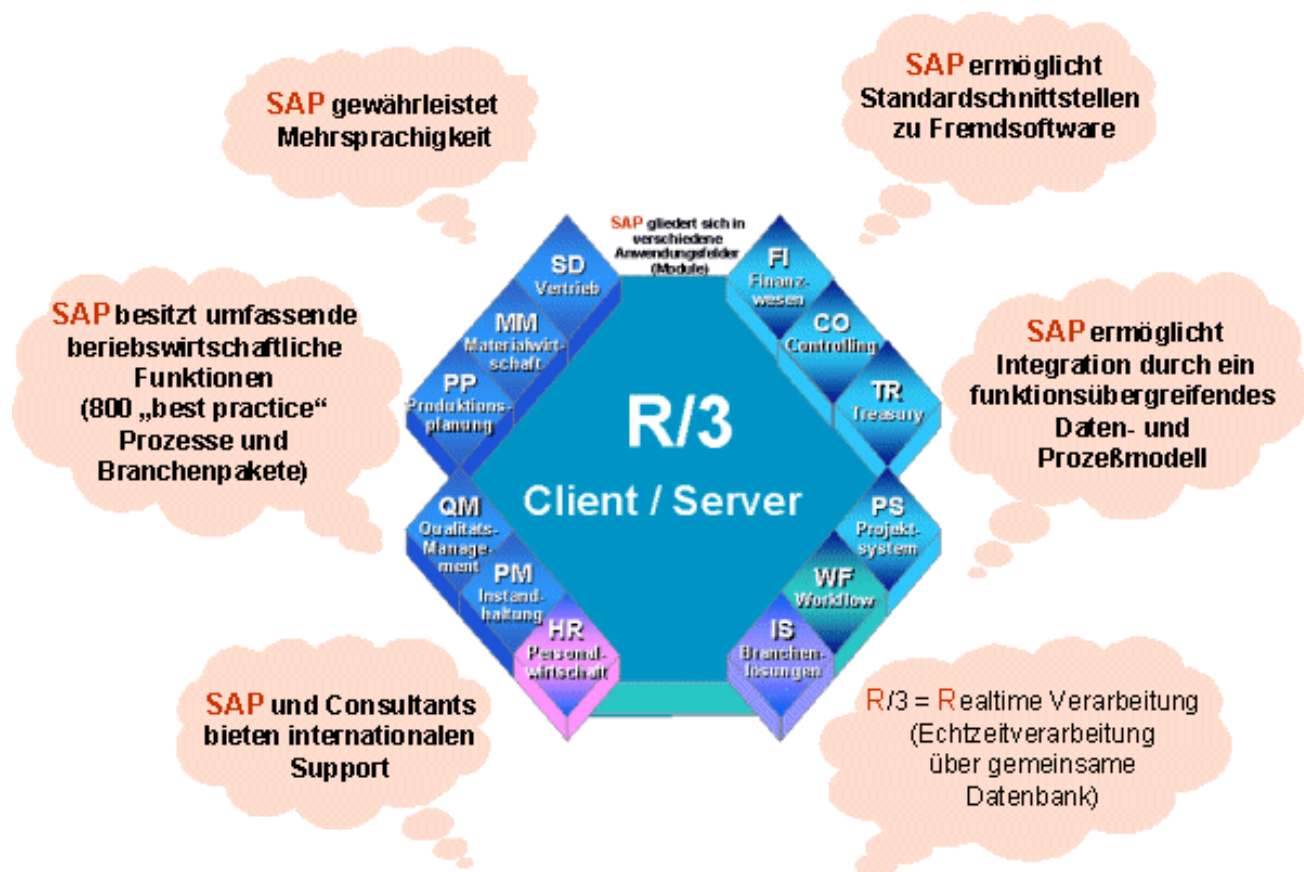


Abbildung 3 – Aufbau und Funktionalität von SAP R/3 [5]

Erforderliche Anpassungen der Standardmodule an Kundenwünsche und kundenspezifische Gegebenheiten werden über ein in die Software integriertes Customizing\*-System abgewickelt. Für die Erstellung von Ergänzungen der Standardfunktionen bzw. für Add-on-Entwicklung wird eine komplette Softwareentwicklungsumgebung angeboten, mit deren Hilfe auch individuelle Lösungen integriert eingebunden werden können.

Desweiteren wird eine eigene bereitgestellt, in der eigene Berichte und Auswertungen oder zusätzliche Anwendungen erstellt werden können.

Das R/3-System unterteilt sich in 3 Ebenen:

- Präsentation
- Anwendung
- Datenbank

Die Präsentationsebene bildet die Schnittstelle zwischen dem R/3-System und dem Benutzer. Mit Hilfe des SAPGUI\* werden in der Präsentationsebene die Anwendungsdaten aufbereitet und auf der grafischen Benutzeroberfläche dargestellt. Das SAPGUI orientiert sich an ergonomischen Gestaltungsnormen, wie dem Windows Style Guide. Durch die Unabhängigkeit des SAPGUI von der Hardware wird erreicht, daß sich alle grafischen Elemente dem Benutzer immer an gleicher Stelle mit derselben Funktionalität darstellen. Ein weiterer Vorteil des SAPGUI ist, daß die grafische Darstellung der Oberfläche erst auf dem Frontend selber erstellt wird. Zwischen Anwendungs- und Präsentationsserver werden keine komplett aufbereiteten Screens verschickt, sondern nur plattformunabhängige Beschreibungen der Darstellungen von ca. ein bis zwei KB pro Bildwechsel. Damit können viele Benutzer gleichzeitig in einem Netz arbeiten, ohne dieses zu überlasten. In der Anwendungsebene liegt die betriebswirtschaftliche Logik des R/3-Systems. Sie ist verantwortlich für den kompletten Ablauf der Funktionen. Benutzeranfragen werden vom Dispatcher an die entsprechenden Workprozesse weitergeleitet und bearbeitet. Das Ergebnis der Verarbeitung wird dann an die Präsentationsebene zurückgereicht.

In der Datenbankebene werden alle vom R/3-System benötigten Daten in einer zentralen relationalen Datenbank gespeichert und verwaltet, d.h. hinzugefügt, aktualisiert und gelöscht. Der Zugriff kann direkt von der Anwendungsebene über das datenbankspezifische SQL erfolgen.

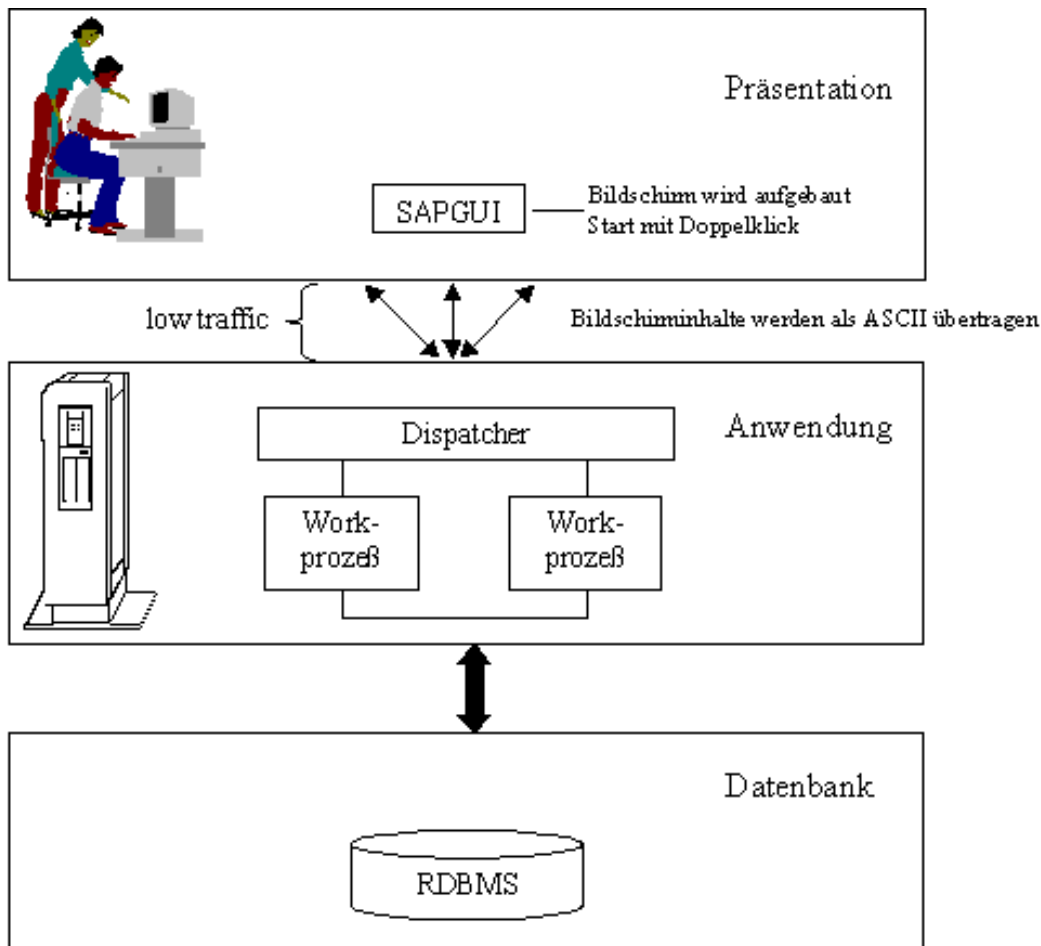


Abbildung 4 - Die 3 Ebenen des R/3-Systems [6]

Die logische Dreiteilung in Präsentation, Anwendung und Datenhaltung findet sich auch in der physischen Struktur wieder. Das R/3-System läßt sich jedoch sowohl als zentrales System auf einem Rechner als auch als zwei- oder dreistufige Architektur aufbauen.

Das gängigste Modell wird kurz vorgestellt:

In der zweistufigen Architektur werden die Terminals durch einen separaten Rechner, den Präsentationsserver, verwaltet. Auf diesem PC- oder X-Terminal-Server läuft das SAPGUI zur Realisierung der Ein- und Ausgabefunktionalitäten des R/3-Systems. Anwendung und Datenbank werden auf einem Server zusammengefaßt.

Besondere Merkmale sind die konsequente Integration und die umfangreiche Parametrisierung, die eine flexible Anpassung an firmenspezifische Anforderungen erlauben.

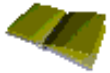
Die elementaren Leistungsmerkmale der R/3-Software sind:

- Die *Funktionalität*, die für SAP-Systeme in über 20 Jahren entwickelt wurde, ist umfassend. Das Finanzwesen profitiert von den Funktionen anderer Anwendungsbereiche wie Vertrieb oder Verkauf, von denen es seinen Buchungsstoff bezieht.
- R/3 bietet nicht nur viele *unterschiedliche Sprachversionen*, sondern auch landesspezifische Funktionen. denn ein Kontenplan oder das Berichtswesen kann z.B. in Frankreich nicht nach gleichen Verfahren wie in den USA oder Rußland angewendet werden.
- Die *Branchenneutralität* zeigt sich an der Liste der Referenzkunden, in der Unternehmen aus unterschiedlichsten Branchen aufgeführt sind. Der Kundenkreis erstreckt sich von der chemischen Industrie und der Automobilindustrie über Handelsunternehmen bis hin zu Banken und Versicherungen. Darüber hinaus kann es bei großen Konzernen genauso wie bei mittelständigen Firmen eingesetzt werden.
- Die Anpassung des R/3-Systems an branchenspezifische und firmenspezifische Besonderheiten erfolgt mit Hilfe von *dialoggesteuerten Customizingfunktionen*. Customizing bedeutet die Vornahme von betriebsspezifischen Standardvorgaben und Verarbeitungsregeln in der Standardsoftware. Werden

beispielsweise für das Mahnverfahren firmenspezifische Verzugstage, Mahntexte oder Berechnungen von Fälligkeitszinsen bestimmt, so wird dieser Vorgang als Customizing bezeichnet. Auch nach einer R/3-Einführung können komfortabel Softwareanpassungen mit Hilfe des Customizing durchgeführt werden, wenn dies aufgrund veränderter Abläufe oder Organisationsstrukturen erforderlich wird, sofern Kontinuität in der Entwicklungsschiene gewährleistet ist.

- Die *einmalige* Speicherung von Daten innerhalb eines SAP-Systems macht Schnittstellen zum Datenaustausch überflüssig, die bei der Verknüpfung von Softwareprodukten unterschiedlicher Hersteller erstellt und gewartet werden müßten. Mehrere Softwaresysteme unterschiedlicher Hersteller verursachen zusätzlich Mehrfachspeicherung; dies wiederum birgt das Risiko unabgestimmter und nicht aktueller Datenbestände sowie darauf aufbauend fehlerhafter Auswertungen.

Einzige Ausnahme sind verteilte Systeme, da dort die Daten erst bis zum zentralen Speicher transportiert werden müssen.



## 4.2 Der Baustein FI (Finanzwesen)

Das Modul FI erfüllt die internationalen Anforderungen, die an ein externes Rechnungswesen einer Unternehmung gestellt werden: Es verfügt über einen offenen, integrierten Datenfluß und vereinfacht die finanzwirtschaftliche Entscheidungsfindung in hohem Maße.

Im R/3-System finden zwischen den verschiedenen Anwendungen alle Buchungen in Echtzeit statt, z.B. zieht eine Buchung im Nebenbuch "Kreditoren" automatisch eine Gegenbuchung im Hauptbuch nach sich.

Das R/3-Finanzwesen läßt sich folgendermaßen untergliedern:

- Hauptbuchhaltung - die Buchhaltung, für die Darstellung von Werten, auf deren Basis die Bilanz sowie die Gewinn- und Verlustrechnung erstellt werden.
- Konsolidierung – die Zusammenführung der Einzelabschlüsse der einbezogenen Konzerngesellschaften zum Konzernabschluß unter dem Gesichtspunkt der Einheitstheorie.
- Kreditorenbuchhaltung – Buchhaltung für alle Buchungen, die Geschäftspartner betreffen, gegenüber dem Verbindlichkeiten für erhaltene Leistungen bestehen.
- Debitorenbuchhaltung – Buchhaltung für alle Buchungen, die Geschäftspartner betreffen, gegenüber dem Forderungen für erbrachte Leistungen bestehen.
- Anlagenbuchhaltung - ist eine Nebenbuchhaltung der Finanzbuchhaltung in der die betriebswirtschaftlichen Vorgänge zum Anlagevermögen erfaßt werden.
- Spezielle Ledger - ist ein Hauptbuch, basierend auf beliebigen Elementen des Kontierungsblocks. Damit können beliebige kundenorientierte Auswertungen gemacht werden

Das Modul FI stellt eine zentrale Komponente des SAP-Systems dar. Ohne Verwendung von störanfälligen, kostenintensiven und zeitverschiebenden Schnittstellenprogrammen werden Datenbestände von und zu der Finanzbuchhaltung transportiert. Die Komponente FI im modularen Aufbau des Systems R/3 erlaubt einen separaten Einsatz. Die Verwirklichung des Integrationsgedankens erfolgt allerdings erst bei einer Kombination mit anderen SAP-Komponenten.

Das Modul FI ist die Zusammenfassung von Komponenten, die inhaltlich dem externen Rechnungswesen zuzuordnen sind.

Der Leistungsumfang des Moduls FI reicht über allgemeine Funktionen wie Debitoren-, Kreditoren- und Sachkontenbuchhaltung hinaus. Die Anlagenbuchhaltung ist vollständig integriert. Die Finanzdisposition läßt Liquiditätsvorschau, kurzfristige Finanzplanung, aber auch Soll/Ist-Vergleiche zu. Das Finanzinformationssystem übernimmt Funktionen wie Kundenkreditkontrolle, Umsatzanalyse, Konzernberichte und Bilanz/GuV-Analysen. Der Teilbereich Konsolidierung ermöglicht die Erstellung eines Konzernberichtswesens, Währungsrechnung,

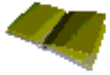
Kapital- bzw. Schuldenkonsolidierung sowie Binnenumsatzermittlung.

Eine Buchführung ist nach dem HGB\* ordnungsgemäß, wenn sie den gesetzlichen Vorschriften entspricht und jederzeit einen Überblick über

- Geschäftsvorfälle,
- Vermögens-,
- Finanz- und
- Erfolgslage

der Unternehmung gestattet.

Das SAP-System erfüllt die Forderung nach Einhaltung der Grundsätze ordnungsgemäßer Buchführung. In integrierten SAP-Anwendungen werden viele Buchungen nicht direkt im Hauptbuch gebucht, sondern aus den Buchungen in verschiedenen Nebenbüchern automatisch abgeleitet. Das komplette System folgt dem Belegprinzip und der Belegsteuerung.



#### 4.2.1 Belegprinzip und Belegsteuerung

Kernpunkt des Belegprinzips ist, daß jeder Geschäftsvorfall in einem eigenen Buchungsbeleg erfaßt und dokumentiert wird. Das Buchen ist nur bei Saldo-Null möglich, von Ausnahmen wie statistischen Belegen (Anzahlungsanforderungen) abgesehen.

Alle Belege in R/3 haben einen einheitlichen Aufbau. Ein Beleg besteht immer aus einem Belegkopf und beliebig maximal 999 Belegpositionen.

Bis zur Version 4 implementierte die SAP-Logik, daß zu jedem Belegkopf mindestens zwei Belegposition vorhanden sein müssen. Ab der Version 4 ist dies nicht mehr zwingend erforderlich. So kann es ab dieser Version vorkommen, daß zu einem Belegkopf keinerlei Positionen vorhanden sind. Die Logik, die dem zugrunde liegt, zeigt sich bei ausgeglichenen Belegen. So gab es bisher einen unausgeglichenen Beleg, d.h. ein Kopf mit einer Position. Dazu kam später ein Ausgleichsbeleg, d.h. ein weiterer Kopf, der auf den ersten Beleg verwies, mit der dazugehörigen Ausgleichsposition. In der neuen Version erkennt SAP, daß hier ein Ausgleich stattgefunden hat, und löscht die Positionen. So bleibt nur der Belegkopf stehen.

Im Belegkopf, aber auch in den Belegpositionen, gibt es eine Reihe von Eingaben, die bei der Belegerfassung immer vorzunehmen sind.

Der Belegkopf beinhaltet alle Daten, die für den gesamten Beleg gelten:

Zu den wesentlichen Angaben im Belegkopf gehört die Belegart, die im Customizing ausführlich definiert wird. Die eindeutige Vergabe einer Belegnummer innerhalb eines Buchungskreises ist selbstverständlich Bestandteil der ordnungsgemäßen R/3-Finanzbuchhaltung.

Der betriebswirtschaftliche Vorgang, z.B. Bestellung, Rechnung, o.ä., bestimmt, welche spezifischen Inhalte für die Belegpositionen relevant sind. Die Steuerung einer Belegposition wird wesentlich durch den Buchungsschlüssel bestimmt.



**Belegkopf: Buchungskreis 0001**

Belegart: **SA** Sachkontenbeleg

Belegkopftext: **D0154000014100000067IMYAL**

Referenz: ..... Belegdatum: **12.04.1999**

Währung: **JOD** / **DEM** Buchungsdatum: **12.04.1999**

Umrechnungskurs: **2,32773** Buchungsperiode: **04 / 1999**

Umrechnungsdat: **12.04.1999**

Bezeichnung: **BKPF** Buchhaltungsbeleg

Objektschlüssel: **410000006700011999** **D01M540**

Erfasser: **IMYAL** Transaktion: **FB01**

Erfassungsdatum: **12.04.1999** Erfaßt um: **11:07:40**

Geändert am: ..... Ltzt. Fortsch: .....

Workflow: ☐ Mikrofilmnr: .....

**Weiter**

**Beleg anzeigen: Übersicht**

Beleg Bearbeiten Springen Zusätze Einstellung Umfeld System Hilfe

✓ [Dropdown] [Icons] [?] [Icons]

Auswählen Belegkopf Währung

Belegnummer: **4100000067** Buchungskreis: **0001** Geschäftsjahr: **1999**

Belegdatum: **12.04.1999** Buchungsdatum: **12.04.1999** Periode: **4**

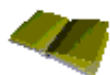
Referenz: ..... Übergreifd.Nr: .....

Währung: **JOD** Texte vorhanden: ☐

Positionen in Belegwährung:

Pos	BS	Konto	Bezeichnung	Kostenst.	Auftrag	ST Betrag in	JOD
001	40	30110	Gebäuden Zugänge			123.456,000	
002	50	30180	WB Gebäude Umb. (			123.456,000-	

Abbildung 5 - Zusammenspiel Kopf mit Position



#### 4.2.2 Organisationsstrukturen im Modul FI

Die Hierarchie der SAP-Ordnungsbegriffe innerhalb eines Mandanten (s.u.) hängt von der Auslegung der R/3-Anwendung ab, also davon, welche R/3-Module tatsächlich im Einsatz sind. Da es aus diesem Grund mehrere Organisationsstrukturen gibt, wird hier die allgemeine R/3-Finanzbuchhaltungstruktur vorgestellt:

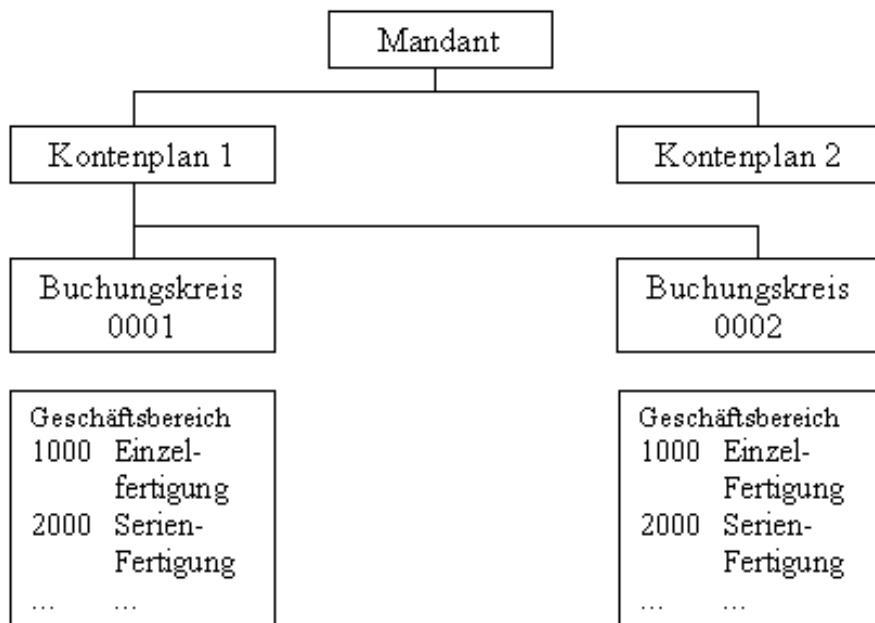


Abbildung 6 - Organisation FI - Grundstruktur [10]

- **Mandant:** Ein Mandant ist der oberste Begriff in der Hierarchie von Abrechnungsbegriffen. Alles, was auf Mandantenebene festgelegt wird, gilt automatisch für alle untergeordneten SAP-Abrechnungsbegriffe, insbesondere den Buchungskreis. Alles, was unter dem Begriff Mandant festgelegt wird, wird mit der Abrechnung eines Konzerns verbunden.
- **Kontenplan** – der Kontenplan ist das zentrale Element der Buchhaltung und erzwingt die Buchung aller FI-relevanten Belege gemäß einer einheitlichen Kontenstruktur. Im Kontenplan sind alle Sachkontenstammsätze der Klasse 0 bis 9 systematisch gegliedert und für jedes Sachkonto\* Kontonummer, Kontenbezeichnung und steuernde Informationen abgelegt. Hierdurch wird die Konsistenz der Mengen- und Werteflüsse innerhalb der Geschäftsprozesse gewährleistet. Innerhalb eines Mandanten können verschiedene Kontenpläne verwaltet werden. Jedem Buchungskreis muß jedoch eindeutig ein im Mandanten definierter Kontenplan zugewiesen werden, wobei ein Kontenplan für mehrere Buchungskreise verwendet werden kann. Kontenpläne werden z.B. nach branchenspezifischen Gesichtspunkten aufgebaut. So können Buchungskreise, die gleichen Branchen angehören, dem gleichen Kontenschema zugeordnet werden.
- **Buchungskreis:** Ein Buchungskreis stellt eine rechtlich selbständige, gemäß den gesetzlichen Vorschriften bilanzierende Einheit innerhalb eines Mandanten dar. Der Buchungskreis bildet die Datenbasis zur Erstellung der Bilanz und der GuV-Rechnung.

Innerhalb eines Mandanten können bis zu 9999 Buchungskreise eingerichtet werden.

Firmenspezifische Festlegungen werden auf der Ebene des Buchungskreises getroffen. Hierzu zählen die Definitionen des Geschäftsjahres, die Hauswährung, Parameter zur Steuerung automatischer Verfahren, wie z.B. Zahlungen oder Mahnungen, und die Zuordnung zu einem definierten Kontenplan.

Firmenspezifische Daten werden hier gespeichert.

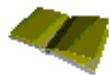
Firmenübergreifende Daten werden auf Mandantenebene angelegt und gespeichert.

Jeder Geschäftsvorfall in der Buchhaltung wird buchungskreisspezifisch erfaßt und ist entsprechend auswertbar.

Mandant und Buchungskreis stehen im Verhältnis 1:n

- **Geschäftsbereich:** Ein Geschäftsbereich ist als wirtschaftlich gesondert zu betrachtende Einheit zu verstehen. Als Geschäftsbereich können z.B. Sparten, Betriebe oder Abteilungen abgebildet werden. Es ergibt sich somit die Möglichkeit, nicht nur über den gesamten Buchungskreis, sondern auch je Geschäftsbereich eine interne Bilanz und GuV-Rechnung zu erstellen, die allerdings nicht alle Anforderungen an offenzulegende Abschlüsse erfüllt. Bei der Erfassung von Geschäftsvorfällen ist der Geschäftsbereich anzugeben oder kann z.T. auch aus anderen Angaben (Werk, Kostenstelle) automatisch abgeleitet werden.





### 4.2.3 Integration mit anderen SAP-Anwendungen

Die Finanzbuchhaltung ist ein zentrales Modul im Integrationsmodell der SAP. Sie wird automatisch mit Daten aus angrenzenden Modulen versorgt und gibt in der Buchhaltung erfaßte Daten ab.

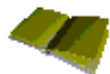
Beispiele für eingehende Daten:

- Ausgangsrechnung aus der Fakturierung (Modul SD)
- Wareneingangswerte aus der Materialwirtschaft (Modul MM)
- Eingangsrechnungen aus der Rechnungsprüfung (Modul MM)
- Verbrauchsdaten aus der Materialwirtschaft (Modul MM)
- Anlagebewegungen aus der Anlagenbuchhaltung (Modul FI-AA)
- Lohn- und Gehaltsbelege aus der Personalwirtschaft (Modul HR)

Beispiele für ausgehende Daten:

- Mit Kostenstelle erfaßte Belege an die Kostenstellenrechnung (Modul CO)
- Mit Projektkontierung erfaßte Belege an die Projektabrechnung (Modul PS)

Nach der Installation der oben genannten Anwendungsbausteine erfolgt der beschriebene Datenaustausch automatisch und aktuell über interne Kommunikationsschnittstellen.

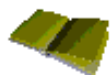


### 4.2.4 Bearbeiten von Belegen im Modul FI

SAP orientiert sich konsequent am Belegprinzip. Das R/3-System läßt nur das Buchen vollständiger Belege zu, d.h. in jedem Beleg müssen die Kopfdaten gefüllt und mindestens eine Soll- und Habenbuchung erfaßt sein. Das Saldo aus Soll- und Habenpositionen muß immer Null ergeben, um den Beleg zu buchen. Desweiteren müssen alle eingegebenen Daten konsistent zum Gesamtsystem sein. Beispielsweise prüft das System, ob Kontierungen erlaubt sind oder für den Beleg notwendige Informationen, wie die Eingabe von Werten für integrierte Module, z.B. einer Kostenstelle (Modul CO), vorhanden sind.

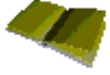
Das Belegprinzip bringt Vorteile beim Zugriff auf Daten, indem dem Anwender nur vollständige Belege zur Bearbeitung angeboten werden. Beim Aufrufen von Belegen kann der Anwender die Vorteile des Belegsystems nutzen, indem er auf eine bestimmte Position zugreift und ihm automatisch weitere Informationen, die z.B. den Kopf oder andere Positionen betreffen, zur Verfügung gestellt werden.

Die zentrale Funktion der Finanzbuchhaltung unter SAP R/3 ist die Transaktion\* <FB01> (Buchen Beleg). Diese Transaktion findet sowohl im Hauptbuch als auch in der Debitoren- bzw. Kreditorenhaltung Verwendung. Die Transaktionen in den verschiedenen Bereichen unterscheiden sich nur durch gewisse Parametervoreinstellungen, wie z.B. Belegart oder Buchungsschlüssel. Da diese Vorschlagswerte überschreibbar sind, kann vom Grundsatz her in jedem Bereich jegliche Art von Buchungsbeleg erfaßt werden.



## 4.2.5 Reporting

Zur Auswertung von Stamm- und Bewegungsdaten werden im SAP-Standardsystem verschiedene Reports ausgeliefert. Einige dieser Auswertungen sind in die Menüstruktur eingebunden. Weitere FI-Reports können über die allgemeinen Transaktionen zum Reporting durch Auswahl des Programmkataloges selektiert werden. Sollten die Standardauswertungen nicht ausreichend oder bestimmte Zusatzinformationen gewünscht sein, können diese Programme mit Mitteln der SAP-Entwicklungsumgebung erstellt werden.



# Input-Verfahren

## Direct Input

Zweck

Ablauf

Technik

## Batch-Input

Zweck

Funktionalität

Technik

Mappen erzeugen

Berechtigungen

Mappen automatisch abspielen

Mappen auswählen

Mappenübersicht

Mappen sortieren und Status anzeigen

Mappen abspielen

Bearbeitungsarten

Mappen korrigieren

Bearbeitungsart: Sichtbar abspielen

Mappen löschen

Mappen sperren und entsperren

Mappenprotokoll anzeigen

Mappen analysieren

Protokolldatei reorganisieren

Buchen

Protokolldateiproblematik

---

Input-Verfahren dienen dazu, Daten aus Fremdsystemen oder SAP-Systemen anderer Arbeitsfelder in ein SAP-System einzustellen. Um aber Einträge in allen notwendigen Tabellen des SAP-Systems zu erzeugen, müssen die Daten so eingespielt werden, als ob sie von einem Anwender tatsächlich gebucht werden. Dieses kann einerseits über bloßes Weiterleiten der Daten an die Buchungsreports geschehen, andererseits aber auch über das automatische Füllen von Bildschirmmasken (Abspielen von Transaktionen).

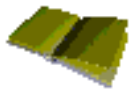
Diese Verfahren existieren für viele Module des SAP-Systems, so z.B. für das Modul FI.

Durch das Aufnehmen von "Fremddaten" in das SAP-System über die Schnittstelle Batch-Input können sehr viele Felder manuell geändert werden.

In der Regel werden Daten vom Benutzer interaktiv, d.h. in einer Dialoganwendung, in das SAP-System eingegeben. Mit besonderen Verfahren werden jedoch bei besonders großen Datenvolumen die Bestände dialogfrei ins SAP-System übernommen, um aufwendige manuelle Erfassung zu umgehen. Diese Möglichkeit der Datenübernahme wird häufig bei der Erstinstallation eines R/3-Moduls und der Übernahme von Daten aus Altsystemen oder beim Datenaustausch mit externen Systemen im laufenden Betrieb angewendet.

Für diesen Zweck bietet das SAP-System spezielle Datenübernahmeprogramme, die es ermöglichen, Daten transaktionsorientiert ins System einzuspielen.

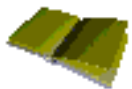
Im Folgenden werden zwei Arten von Input-Verfahren beschrieben. Beide sind möglich, um Fremddaten im SAP-System zu buchen, allerdings haben sie unterschiedliche Funktionalität.



## 5.1 Direct Input

### 5.1.1 Zweck

Der Zweck des Direct-Input besteht darin, Daten mit geringem Aufwand in die entsprechenden Tabellen zu schreiben. Es werden lediglich die Konsistenzprüfungen durch von SAP erstellte Funktionsbausteine vorgenommen.

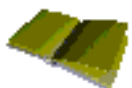


### 5.1.2 Ablauf

Bei diesem Verfahren werden Daten wie beim Füllen von Bildschirmmasken gebucht, allerdings durchlaufen sie nicht die Dynpro\*-Prüfungen, d.h. Kontrollstrukturen (PAI\*, PBO\*) greifen nicht. Es wird auf die Funktionsbausteine zugegriffen, die für die Verbuchung relevant sind.

Folge davon ist, daß einige Fehler unter Umständen nicht sauber abgefangen werden.

Weiterer und gravierender Nachteil des Direct-Input-Verfahrens ist die mangelnde Fähigkeit, einen Workflow zu starten. Dies ist ein gravierender Nachteil, da Unternehmen immer mehr dazu übergehen, Prozeßabläufe zu automatisieren um die Durchlaufzeit nicht mehr von manuellem Weiterleiten von Informationen abhängig zu machen.



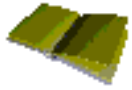
### 5.1.3 Technik

Wenn Daten mit Direct-Input übernommen werden sollen, müssen sie im internen Format aufbereitet sein.

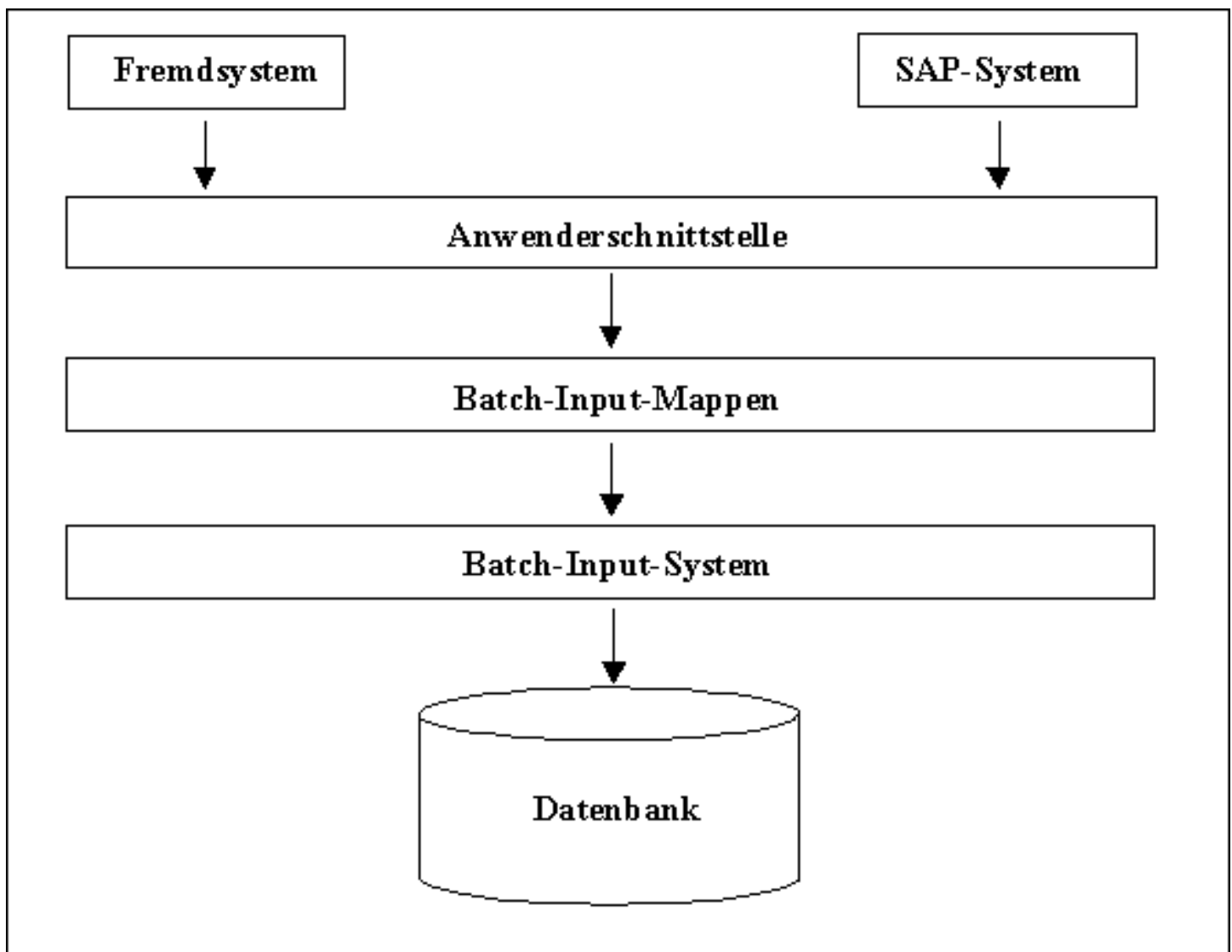
Bei Direct-Input werden keine Dynpros prozessiert, sondern die Belege durch den Aufruf von Funktionsbausteinen direkt gebucht. Direct-Input steht nur für eine Teilmenge der Buchungsmöglichkeiten zur Verfügung. Es können keine <FB05>-Buchungen (Buchen mit Ausgleichen) und keine Buchungen mit Sonderhauptbuchvorgängen\* erzeugt werden.

Durch das Ignorieren der Dynpro-Prüfungen hat das Verwenden von Direct-Input einen ähnlichen Charakter wie das Verwenden des KTW: es wird "hart" in Tabellen geschrieben, ohne das Durchlaufen von Dynpro-Plausibilitätsprüfungen.

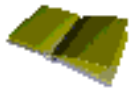
Für das Modul FI wird der Direct Input durch den Report RFBIBL00 mit dem Parameter 'D' angestoßen. Über den Funktionsbaustein AC\_DOCUMENT\_DIRECT\_INPUT werden die Belege in die Tabellen BKPF und BSEG gebucht. Bei Auftreten eines Buchungsfehlers oder Auftreten eines Plausibilitätsfehlers wird automatisch eine BI-Mappe erzeugt, die dann wie beim normalen Batch-Input über die Transaktion <SM35> abgespielt wird.



## 5.2 Batch-Input

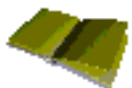


Mit Hilfe der Batch Data Communication (Batch-Input) wird in der Hintergrundverarbeitung ein Benutzerdialog simuliert, der die gleichen Dynpros und Transaktionen nutzt wie ein Anwender, der die Daten manuell im Dialog eingibt. Der Anwender bekommt vollständig gefüllte Dynpros angeboten, die er dann buchen kann. Desweiteren werden BI-Mappen\* den üblichen Benutzerberechtigungsprüfungen durch das System unterzogen.



### 5.2.1 Zweck

Durch dieses Verfahren wird sichergestellt, daß die Daten bei der Batch-Einspielung die gleichen Prüfungen durchlaufen, wie bei der manuellen Dialogeingabe.



### 5.2.2 Funktionalität

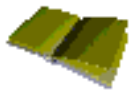
Mit den Funktionen der Batch-Input-Verwaltung kann man folgende Aufgaben erledigen:

- BI-Mappen zur Eingabe von Daten in das SAP-System abspielen.
- Für die Fehlersuche stellt die BI-Schnittstelle zwei Modi zum interaktiven Abspielen einer Mappe zur Verfügung.
- Das Hintergrundprogramm einplanen, das Mappen für die Hintergrundverarbeitung auto-matisch startet.
- Mappen bis zu einem bestimmten Datum sperren. Solange eine Mappe gesperrt ist, kann sie nicht abgespielt werden.
- BI-Mappen löschen.
- Mappen nach einem Abbruch aufgrund eines Systemfehlers oder einer Systemabschaltung beim Erstellen oder Abspielen für nochmaliges Abspielen freigeben.
- BI-Mappen vor oder nach dem Abspielen analysieren. Mit der Analysefunktion kann man die Eingabedaten der Transaktionen in einer Mappe anzeigen. Man hat dabei folgende Möglichkeiten:
  - Daten auf den Bildschirmbildern anzeigen, die durch die Mappe aufgerufen wurden.
  - Listen von Eingabefeldern nach Feldnamen mit den jeweiligen Mappendaten anzeigen.
  - Protokolle von BI-Mappen anzeigen.



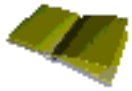
### 5.2.3 Technik

Realisiert wird dies durch spezielle ABAP/4-Programme, die Batch-Input-Mappen erzeugen. In diesen Mappen werden alle Transaktionen und Daten gesammelt, die zur Einspielung ins System notwendig sind. Durch die Mappenerstellung selber werden noch keinerlei Stamm\*- bzw. Bewegungsdaten\* im SAP-System angelegt. Die Verbuchung der einzelnen Transaktionen erfolgt mit dem erfolgreichen Abspielen der Mappen.



#### 5.2.3.1 Mappen erzeugen

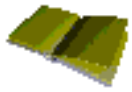
Für die Datenübertragung verwendet das Ausgangssystem eine Übertragungsschnittstelle, die von einem R/3-Anwendungsprogramm im Zielsystem zur Verfügung gestellt wird. Das Schnittstellenprogramm der Anwendung erzeugt dann eine Batch-Input-Mappe. Hierbei handelt es sich um ein ABAP/4-Programm, das die Transaktionsaufrufe und Daten zusammenstellt, aus denen eine Mappe besteht. Wenn die Batch-Input-Mappe Daten aus einer externen Quelle enthält, bereitet das Programm diese Daten auf, um den Anforderungen der entsprechenden Eingabefelder gerecht zu werden. In der Regel werden solche Programme von den R/3-Anwendungen zur Verfügung gestellt.



### 5.2.3.2 Berechtigungen

Beim Erstellen einer Mappe werden ein Mandant und ein Benutzer mit ihr assoziiert. Wenn die Mappe im Hintergrund abgespielt wird, führt das System während des Abspielens die Berechtigungsprüfung für diesen Benutzer durch. Die Berechtigungsprüfung ist unabhängig davon, ob man die Mappe selbst gestartet haben oder ob die Mappe durch einen Hintergrund-Job gestartet wird.

Mappen, die man in einem der interaktiven Modi abspielt, unterliegen den jeweiligen Berechtigungen.



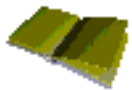
### 5.2.3.3 Mappen automatisch abspielen

In den meisten Fällen werden Mappen nicht interaktiv gestartet, sondern durch einen Hintergrund-Job startet in regelmäßigen Abständen noch nicht abgespielte Mappen. Das interaktive Abspielen von Mappen beschränkt sich normalerweise auf das Testen und Korrigieren von Mappen.

Zum automatischen Starten von Batch-Input-Mappen wird das ABAP/4-Programm RSBDCSUB periodisch ausgeführt. Das Programm plant Mappen zur sofortigen Hintergrundverarbeitung ein.

Zu verarbeitende Mappen können nach folgenden Kriterien ausgewählt werden:

- Mappenname
- Datum und Uhrzeit der Erstellung
- Status: abspielbereit oder fehlerhaft



### 5.2.3.4 Mappen auswählen

Über die Transaktion <SM35> erreicht man das Einstiegsbild des Batch-Input-Systems. Von dort kann man Mappen mit einer beliebigen Kombination der folgenden Kriterien auswählen:

- Mappenname
- Erstellungsdatum (laut Mappenübersicht)
- Mappenstatus

Man kann auch zwischen zwei Aktionen wählen:

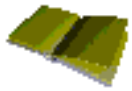


- zur Anzeige der Mappenübersicht wählt man 'Übersicht':

Man kann Mappen in der Übersicht starten, analysieren oder löschen

- zur Anzeige der Mappenkontrolle wählt man 'Protokoll':

Zu jeder Mappe wird beim Abspielen ein Protokoll erstellt. Über die Liste der Mappenprotokolle kann man weitere Aktionen ausführen, wie etwa Mappen analysieren.



### 5.2.3.5 Mappenübersicht

Wichtige Informationen in der Mappenübersicht sind:

- *Datum und Zeit*

Datum und Zeit der Mappenerstellung

- *Gesperrt*

Wenn Mappen gesperrt sind, erscheint in dieser Spalte das gewünschte Datum, an dem das System die Mappe freigibt. Eine gesperrte Mappe kann nicht gestartet werden.

- *Ersteller*

Der Benutzer, der die Mappe erstellt hat.

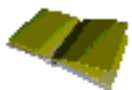
- *Trans und Dynpro*

Die Anzahl der Transaktionen und Bildschirmbilder, die in einer Mappe noch zu verarbeiten sind.

- *Berechtigung*

Der Benutzer, mit dessen Berechtigung eine Mappe abgespielt wird, die für die Hintergrundverarbeitung vorgesehen ist

Eine Statistik zu den Transaktionen einer Mappe erhält man, indem man die gewünschte Stelle markiert und *Statistik ...* wählt



### 5.2.3.6 Mappen sortieren und Status anzeigen

Mappen in der Mappenübersicht sind nach Datum und Uhrzeit ihrer Erstellung sortiert und entsprechend ihrem Status in verschiedene Anzeigen unterteilt.

Nachfolgend werden die verschiedenen Stati dargestellt.

- *Noch zu verarbeitende Mappen*
- Die Spalten Trans und Dynpro in der Mappenübersicht enthalten die Anzahl der Transaktionen und Bilder, die in der jeweiligen Mappe enthalten sind
- *Fehlerhafte Mappen*
- *Fehlerhafte Transaktionen* werden abgebrochen, fehlerfreie hingegen ausgeführt. Die Spalten Trans und Dynpro in der Mappenübersicht enthalten die Anzahl der fehlerhaften Mappen.

Man kann eine Mappe neu starten und die fehlerhaften Transaktionen in einem der Dialogmodi des Batch-Input-Systems korrigieren.

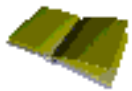
- *Bearbeitete Mappen*

Wenn man Informationen zu einer erfolgreich abgespielten Mappe benötigt, läßt man sich das erstellte Protokoll anzeigen. Zu allen bearbeiteten Mappen ist ein Protokoll vorhanden. Eine bearbeitete Mappe kann kein zweites Mal abgespielt werden.

Mit der Option KEEP abgespielte Mappen bleiben nach erfolgreicher Verarbeitung in der Mappenübersicht stehen. Andere Mappen werden gelöscht.

- *Mappen in Erstellung oder Mappen in Bearbeitung*

Diese Status erscheinen normalerweise nur, wenn zufällig die Mappenübersicht aufgerufen wird, während eine Mappe gerade erstellt wird (d.h. in die Mappenübersicht aufgenommen wird). Sie erscheinen jedoch auch, wenn die Erstellung einer Mappe durch einen Systemfehler unterbrochen wurde.

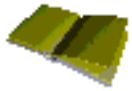


### 5.2.3.7 Mappen abspielen

Beim Abspielen einer Batch-Input-Mappe werden Transaktionen ausgeführt und Daten in ein SAP-System eingespielt.

Zum Abspielen einer Mappe wählt man im Hauptmenü des Batch-Input-Systems den Punkt Übersicht, wählt dann eine Mappe aus, und startet sie mit Abspielen oder Im Hintergrund abspielen.

Man kann eine beliebige Mappe aus der Übersicht der noch zu verarbeitenden Mappen starten, sofern diese nicht gesperrt ist. Über die Funktion Sichtbar abspielen und Nur Fehler anzeigen kann man Transaktionen mit dem Status Fehlerhaft neu starten. Mappen mit dem Status Verarbeitet kann man nicht neu starten.



### 5.2.3.8 Bearbeitungsarten

Man kann zwischen 3 Bearbeitungsarten für Mappen wählen:

- Sichtbar abspielen

fehlerhafte Transaktionen können interaktiv korrigiert und noch nicht ausgeführte Transaktionen stufenweise abgearbeitet werden

- Nur Fehler anzeigen

Diese Bearbeitungsart gleicht der Bearbeitungsart Sichtbar abspielen, außer, daß Transaktionen, die noch nicht ausgeführt wurden und keine Fehler enthalten, nichtinteraktiv ablaufen. Bildschirmbilder, auf denen der Fehler auftreten, werden angezeigt.

- Hintergrund

Bei dieser Bearbeitungsart wird eine Mappe für die sofortige Verarbeitung im Hintergrundsystem eingeplant. Der Bildschirm steht wieder für Eingaben zur Verfügung, sobald die Mappe an das Hinter-grundsystem übergeben wurde. Eine verarbeitete Mappe wird auf eine der drei nachfolgend beschriebenen Weisen behandelt:

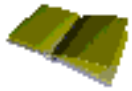
- Das System löscht eine Mappe aus der Mappenübersicht, sobald sie erfolgreich abgespielt wurde. Das Ergebnis kann angezeigt werden, indem die Protokolldatei aufgerufen wird.
- Wenn beim Erstellen einer Mappe die Option KEEP eingestellt war, bleibt der Mappenname in der Mappenübersicht stehen, auch wenn die Mappe erfolgreich verarbeitet wurde. Ihr Status ändert sich zu Bearbeitet.

Die Mappe kann kein zweites Mal abgespielt werden. Außerdem muß sie manuell gelöscht werden, wenn sie nicht mehr benötigt wird.

- Wenn eine Transaktion in der Mappe einen Fehler enthält, wird die Transaktion abgebrochen. Alle übrigen Transaktionen werden abgeschlossen. Die Mappe bleibt in der Mappenübersicht und erhält den Status Fehlerhaft. Es ist möglich, eine Mappe mit einer interaktiven Bearbeitungsart zu korrigieren und die Verarbeitung zu Ende zu führen.

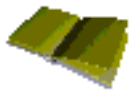
Eine Transaktion enthält einen Fehler, wenn eine Fehlermeldung vom Typ E (Error) oder A (Abbruch) ausgegeben wird. Andere Meldungen werden ignoriert und haben keinen Einfluß auf die Verarbeitung einer Mappe.

Eine Mappe bleibt ebenfalls mit dem Status Fehlerhaft in der Mappenübersicht stehen, wenn die Mappenverarbeitung mit der Eingabe /bend im Befehlsfeld abgebrochen wurde.



### 5.2.3.9 Mappen korrigieren

Beim Abspielen einer Mappe im Hintergrundmodus markiert das System die fehlerhaften Transaktionen. Alle übrigen Transaktionen der Mappe werden abgeschlossen. Die Mappe bleibt in der Mappenübersicht stehen und wird im Abschnitt *Fehlerhafte Mappen* angezeigt. Fehlerhafte Transaktionen können korrigiert und neu abgespielt werden. Auch hier können die Mappen wieder sichtbar abgespielt werden oder nur die Fehler angezeigt werden.



### 5.2.3.10 Bearbeitungsart: Sichtbar abspielen

Hierbei werden die einzelnen Dynpros mit den BI-Daten angezeigt. Wenn der Inhalt den Vorstellungen entspricht, ist die Seite mit der ENTER-Taste zu bestätigen.

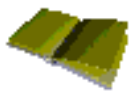
Bereits ausgeführte Transaktionen werden kein zweites Mal ausgeführt.

Das interaktive Abspielen einer Mappe kann mit der Transaktion **/bend** abgebrochen werden. Der Status bekommt den Wert *fehlerhaft* und die Mappe wird in die Mappenübersicht eingestellt.

Beim interaktiven Abspielen einer Mappe kann über den Transaktionscode **/bdel** eine Transaktion unwiderruflich aus der Mappe gelöscht werden.

Wurde die Mappe jedoch mit der Option *KEEP* erstellt, ist zumindestens eine Analyse der gelöschten Transaktion möglich.

Das Löschen einer Transaktion wird im Mappenprotokoll gespeichert. Diese Bearbeitungsart wird auch als "hell abspielen" bezeichnet.

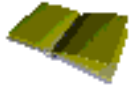


### 5.2.3.11 Mappen löschen

Man kann eine Mappe aus der Mappenübersicht löschen. Dabei werden normalerweise alle Informationen über diese mitgelöscht.

Normalerweise löscht das System eine Mappe nach dem Abspielen automatisch aus der Mappenübersicht. Ist die Mappe jedoch mit der Option *KEEP* erstellt worden, so muß sie manuell gelöscht werden.

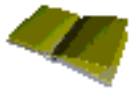
Wird die Mappe manuell gelöscht, kann eingestellt werden, ob das Protokoll mitgelöscht werden soll oder nicht.



#### 5.2.3.12 Mappen sperren und entsperren

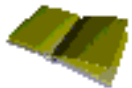
Eine gesperrte Mappe wird erst nach ihrem Sperrdatum abgespielt. Mögliche Gründe, eine Mappe zu sperren, sind:

- eine Mappe soll erst zu einem bestimmten Zeitpunkt verarbeitet werden
- eine fehlerhafte Mappe soll nicht erneut gestartet werden
- eine Mappe, für die beim Abspielen besondere Mittel erforderlich sind, soll nicht unbeabsichtigt abgespielt werden



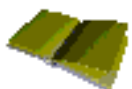
#### 5.2.3.13 Mappenprotokoll anzeigen

Zu jeder BI-Mappe ist ein Protokoll vorhanden, welches sich anzeigen läßt und alle Fehlermeldungen zu Transaktionen in Mappen enthält.



#### 5.2.3.14 Mappen analysieren

Hierbei ist es möglich, den BI-Ablauf zu überprüfen. Mit geeigneten Analysefunktionen kann man vor und nach dem Mappenabspielen arbeiten.



#### 5.2.3.15 Protokolldatei reorganisieren

In periodischen Abständen soll der Report RSBDCREO die Protokolldatei des BI-Systems,

die *BI<SAP-Systemname><Instanzname>* heißt und im zentralen SAP-Verzeichnis "*global*" im Hostsystem gehalten wird, reorganisieren. BSP: Die Log-Datei auf dem F03 (Name des Produktivrechners der BAYER AG, auf dem FI-Belege gebucht werden.) heißt:

**/usr/sap/F03/SYS/global/BIrwwq100**

RSBDCREO minimiert die Größe der Protokolldatei durch Reorganisation.

Beim Löschen einer Mappe wird das zugehörige Protokoll zum Löschen vorgemerkt. Erst bei einer Reorganisation wird dieser Speicherplatz freigegeben.

Ein Protokoll wird nur dann gelöscht, wenn die Mappe, für die das Protokoll erstellt wurde, bereits aus der Mappenübersicht gelöscht wurde.

Der Report hat folgende Funktionen:

1. Löschen von abgespielten Mappen (erfolgreich verarbeitete Mappen) die noch im System sind und deren Protokolle. Nur diese Mappen werden gelöscht, keine '*noch zu verarbeitende Mappen*', keine '*Fehlerhafte Mappen*' etc.
2. Löschen von Protokollen zu denen es keine Mappen mehr gibt.
3. Reorganisation der Protokolldatei (schrumpfen der Dateigröße, wenn Protokolle gelöscht wurden).

Auf den jeweiligen Verzeichnissen muß zur Reorganisationszeit Platz vorhanden sein

Beispiel: Protokolldatei **BIserv00** mit einer Größe von 125 MB. Es werden weitere 125 MB zur Reorganisation dieser Datei benötigt. Die gesamte Datei wird einmal temporär kopiert, das Original gelöscht, und nur diejenigen Protokolle wieder von der Kopie ins Original geschrieben, die kein Löschkennzeichen aufweisen.

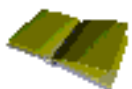
Beim Starten des Reports RSBDCREO sind per Parametereingabe Programmsteuerungen möglich. Da für das Reorganisieren ein Löschen ...

Es können gelöscht werden:

- Mappen, die verarbeitet sind, aber nicht das Protokoll
- Protokolle, von denen es keine Mappe mehr gibt

Reorganisation der BI-Protokolldatei

- Mappen, die verarbeitet sind, mit Protokoll



## 5.2.4 Buchen

Wenn ein Beleg gebucht wird, sieht man als Ergebnis einen Eintrag in einer Tabelle und kann sich den Beleg im Modul FI ansehen. Aber „hinter den Kulissen“ geschieht einiges mehr, was an dieser Stelle angerissen werden soll.

Das R/3-System unterstützt zwei Arten der Datenbankänderung:

- Die Anwendung kann den Dialog-Workprozeß direkt mit der Änderung der Tabelle beauftragen. Hierzu dient der R/3-SQL-Befehl *update*, den ein ABAP/4-Programm verwenden kann.
- Üblicherweise verfügt jede R/3-Transaktion über eine spezielle Funktion zum Sichern. Hierbei veranlaßt der Dialog-Workprozeß mit dem ABAP/4-Sprachelement *call function ... in update task*, daß der Auftrag zur Datenbankänderung an einen speziell dafür vorgesehenen Workprozeß vom Typ *V* übergeben wird. Workprozesse dieser Art nennt man auch Verbucher.

SAP-Transaktionen erstrecken sich im Normalfall über mehr als ein Dynpro. Auf jedem dieser Dynpros kann der Benutzer Daten eingeben, zu anderen Funktionen verzweigen, aber auch ganz bewußt die Funktion <SICHERN> wählen. Es ist aber auch jederzeit möglich, die Funktion <ABBRECHEN> zu wählen.

Das Dialogprogramm weiß zu keinem Moment, auf welchem Bildschirmbild der Benutzer eine der beiden Funktionen wählt. (siehe Abbildung unten)

Statt eines direkten Update-Befehls, der die Datenbankänderung bei jedem Dialogschritt durchführen würde, werden die Änderungsaufträge zunächst mit dem Befehl *call function ... in update task* in der Verbucher-Protokolldatei gesammelt. Hierzu dient die Datenbanktabelle VBLOG.

Da jeder Dialogschritt einen anderen Dialog-Workprozeß verwenden kann, sendet der R/3-Systemkern bei jedem Dialogschritt ein *commit work* an die Datenbank. Hierdurch werden alle Datenbankänderungen für die Datenbank bestätigt. Jeder Update- oder Insert-Befehl wird bei einem Bildwechsel durch diesen SAP-Commit auf der Datenbank abgeschlossen.

Erst bei Ende der Transaktion wird durch den Benutzerbefehl <SICHERN> der SQL-Befehl *commit work* vom Anwendungsprogramm ausgeführt. Dadurch wird der Verbucher-Protokollsatz abgeschlossen, und der Dialog-Workprozeß gibt den Änderungsauftrag an den Verbucher-Workprozeß weiter. Dieser liest dann die Daten der Protokolldatei und führt die Datenbankänderung durch. [2]

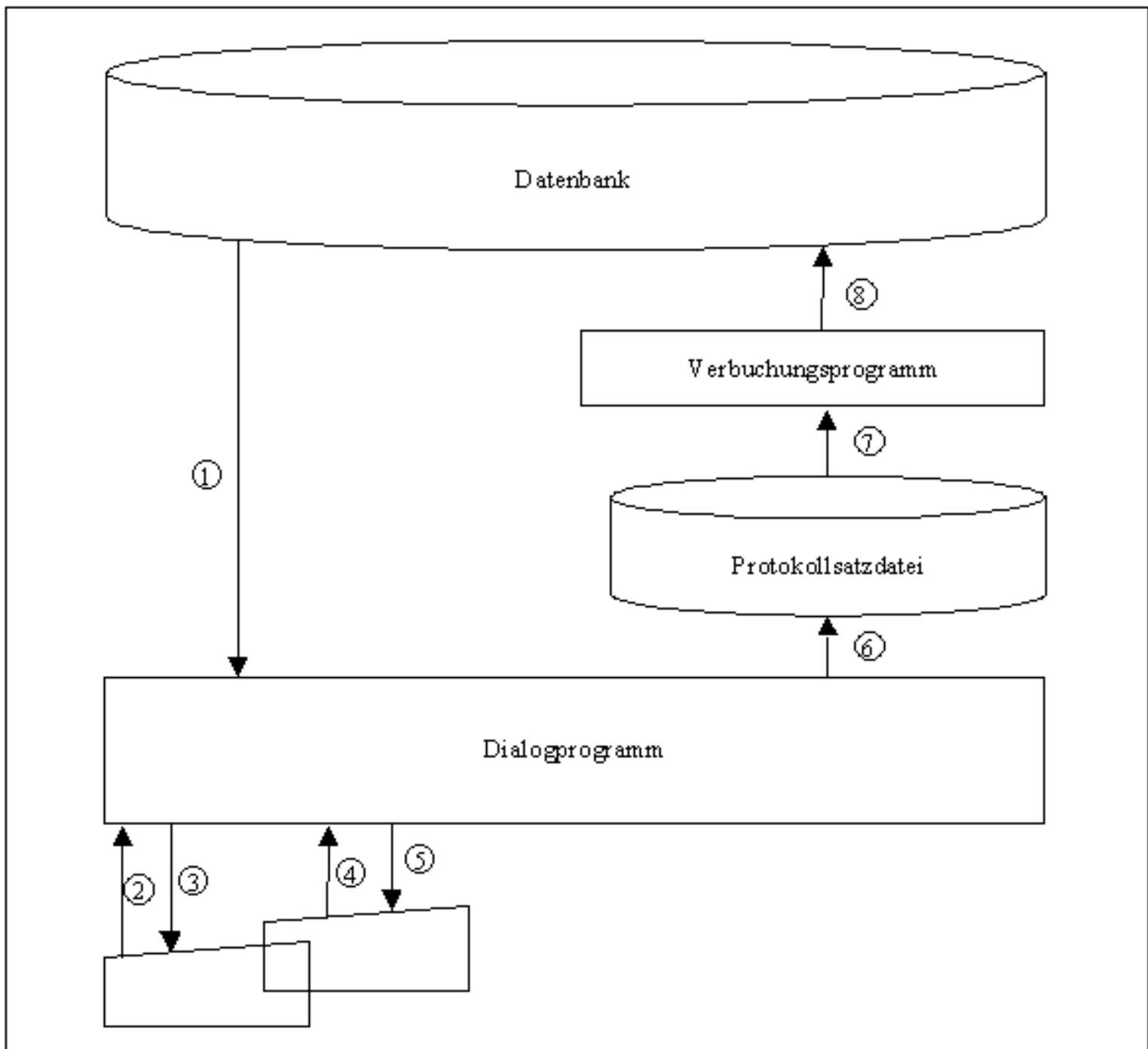
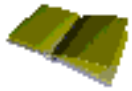


Abbildung 8 - Update-Service [2]

Die Abbildung zeigt den bereits angesprochenen Vorgang des Verbuchens. Im Schritt 1 erhält das Dialogprogramm Daten vom DB-Server. Auf diese Daten kann der Anwender über die Dynpros zugreifen. Die Schritte 2 – 5 zeigen das Zusammenspiel verschiedener Dynpros mit dem Dialogprogramm bei Ablauf einer Transaktion. In Schritt 6 gibt das Dialogprogramm die gesammelten Informationen an die Protokollsatzdatei weiter, die in Schritte 7 über den Verbucher in Schritt 8 die Änderungen auf die Datenbank schreibt.





### 5.2.5 Protokolldateiproblematik

In einem der vorherigen Abschnitte wurde bereits die BI-Protokolldatei angeschnitten. In diesem Kapitel soll etwas näher auf diese Datei und ihre Eigenschaften eingegangen werden. Wie bereits erwähnt, werden in dieser Datei sämtliche Protokolle jemals abgespielter Mappen gesammelt. Diese können dann bei Bedarf abgerufen werden.

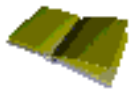
Die Protokollsatzdatei existiert auf jedem SAP R/3-System (auf jedem Application Server) und besitzt auf Produktivsystem schnell Größen von mehreren GB (zum Vergleich: die Datei auf dem Produktivsystem FI der BAYER AG hat die Größe 3 GB).

Diese Datei hat die Eigenschaft, daß sie rein sequentiell ist, und in ihr alle einzelnen Protokolle hintereinander gespeichert sind. Sie ist von der Anwenderseite her nicht auswertbar.

Mit jedem weiteren Abspielen von BI-Mappen werden neue Daten im SAP R/3-System verbucht, und die Protokolldatei wächst weiter. Der im Rahmen dieser Diplomarbeit entwickelte Report übernimmt die Funktion des BI-Mappenprotokolls und weist Abweichungen der Buchungen aus dem MBS-System von den Buchungen im SAP-System aus.

Somit kann das einzelne Mappenprotokoll gelöscht werden. Dies jedoch stellt ein Problem dar, da mit konventionellen Mitteln der genaue Startpunkt der Mappe nicht auffindbar ist. Die Mappenübersicht (Transaktionscode: <SM35>) bietet keine Löschmöglichkeit von Teilen der BI-Protokollsatzdatei.

Beim Löschen einer Mappe wird das zugehörige Protokoll zum Löschen vorgemerkt und bei einer Reorganisation dieser Speicherplatz freigegeben.





Rechtsanwaltskanzlei: der eine darf nichts von dem anderen wissen. Jeder Mandant hat seine Besonderheiten, so auch im Buchungs- und Zahlungsverkehr.

Ein Kunde in Frankreich hat beispielsweise eine andere Steuerrechnung als ein Kunde in Deutschland. Diese Besonderheiten werden durch die Mandantenautonomie berücksichtigt.

Aus dieser Logik heraus existiert in SAP ein sogenanntes Customizing, d.h. Tabellen werden an die Besonderheiten der jeweiligen Mandanten angepaßt. So kann es sein, daß für manche Mandanten bestimmte Felder nicht gefüllt werden dürfen, obwohl diese im Stammsatz vorhanden sind oder vorher definierte Feldwerte werden überschrieben.

Wenn nun Daten aus einem Fremdbuchungssystem in SAP eingespielt werden sollen, müssen alle Buchungsfelder, die sowohl im einen System vorhanden sind, in das andere übertragen werden. Aufgrund der o.g. Mandantenautonomie sind manche Felder in SAP eventuell gar nicht zu füllen.

Die Fragestellung ist nun: Woher bezieht das SAP-System die Kenntnis, welche Felder zu füllen sind?

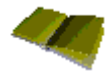
Ergebnis dieser Problemstellung war ein Programm unter SAP, der in der Informatikabteilung der BAYER AG entwickelt wurde, und der als Eingabedatei eine sequentielle Datei auf Betriebssystem-Ebene (UNIX) benutzt. Diese Datei enthält die in SAP zu verbuchenden Belege, aufbereitet gemäß den Konventionen der Batch-Input-Schnittstelle des R/3-Systems.

Dieser Report hat u.a. die Funktion, in einer Customizing-Tabelle „nachzuschauen“, welche Felder bei bestimmter Feldbelegung wie geändert werden müssen. Felder, die „ausgeblendet“ werden, bekommen ein vordefiniertes NO-DATA-Zeichen, im vorliegenden System ist dies das Zeichen „/“.

Die Ausgabedatei des „Ausblenders“, die u.U. bereits anders gefüllte Felder als die ursprüngliche Datei mit den originären MBS-Daten enthält, wird dann in einem letzten Schritt zu BI-Mappen verarbeitet.

Schon aus dieser Logik heraus ist es zu erwarten, daß Feldinhalte der MBS-Belege im SAP-System drastisch verändert sein können.

Das Problem ist das Unterscheiden von Abweichung, die als Folge des Ausblenders anzusehen und damit als unkritisch einzustufen sind, und den Abweichungen, die als Folge von manuellen Benutzereingriffen auftreten.



### 6.1.1 Erstellen und Abspielen von BI-Mappen

Der Report RFBIBL00 erstellt nun auf Basis der Ausgabedatei des „Ausblenders“ Batch-Input-Mappen im SAP R/3-System. Diese können dann gezielt von einem Anwender abgespielt werden.

Das Programm ist in erster Linie dafür gedacht, Batch-Input-Mappen zu erzeugen. Durch den Parameter 'Art der Datenübernahme' können Belege aber auch sofort mit 'Call Transaction' oder 'Direct-Input' erzeugt werden.

Diese beiden Verfahren bringen Performance-Verbesserungen bei der Übernahme von großen Datenmengen (mehr als 10000 Transaktionen).

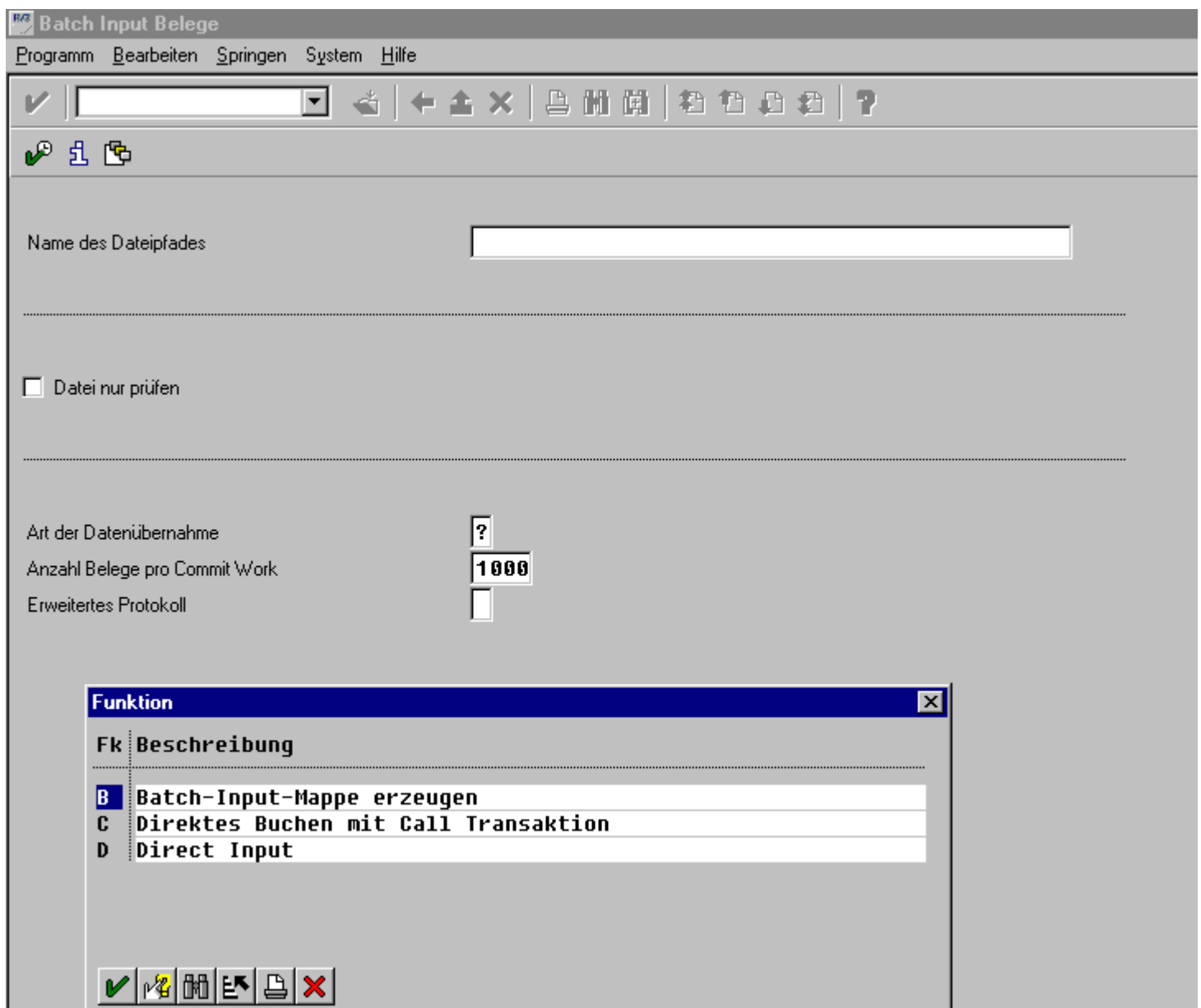


Abbildung 10 – Startbild RFBIBL00

- *Direct-Input*: Hierbei wird keine Batch-Input-Mappen im eigentlichen Sinn erzeugt. Die Belege werden direkt in die erforderlichen Tabellen geschrieben und gebucht. Ist die Buchung in Ordnung, bekommt der Benutzer eine Meldung über die Belegnummer, bei fehlerhaften Buchungen wird eine Fehlermappe erzeugt. Sie enthält nur die Buchungen, die fehlerhaft sind.
- *Direktes Buchen mit Call Transaktion*: Der Unterschied von Call Transaction zu Batch-Input besteht darin, daß die Belege sofort durch das Aufrufen der entsprechenden Transaktionen gebucht werden. Dabei werden die gleichen Dynpros wie bei Batch-Input prozessiert.
- *Batch Input Mappe erzeugen*: Hierbei wird nicht verbucht, sondern es wird „lediglich“ eine BI-Mappe erzeugt. Diese kann zu einem gewünschten Zeitpunkt über die Transaktion SM35 abgespielt werden.

Jede Buchung oder Mappenerzeugung generiert Einträge in die Tabellen APQD (Belegsegmente mit Steuerzeichen) und APQI (Kopf).

Werden nun die BI-Mappen mit den MBS-Daten abgespielt, so kann es sein, daß Feldinhalte zwar im MBS-System eingegeben werden können, beim Buchen in SAP jedoch zu Fehlermeldungen führen. Folge dessen sind Belege, die sich mit den aktuellen Einstellungen nicht buchen lassen. Die Eingabeprüfungen des MBS-Systems sind nicht so streng wie die im SAP-System, da das SAP-System die eingegebenen Daten evtl. in anderen Bausteinen weiterverwendet, MBS als eigenständig gekapseltes System jedoch nicht.

Resultat dieses Problems ist, daß entweder die komplette Transaktion ersatzlos gestrichen wird, oder über Umwege gebucht wird. In diesem Fall ist es jedoch unmöglich, den Zusammenhang zwischen dem ursprünglichen MBS-Beleg und dem SAP-Beleg nachzuweisen.

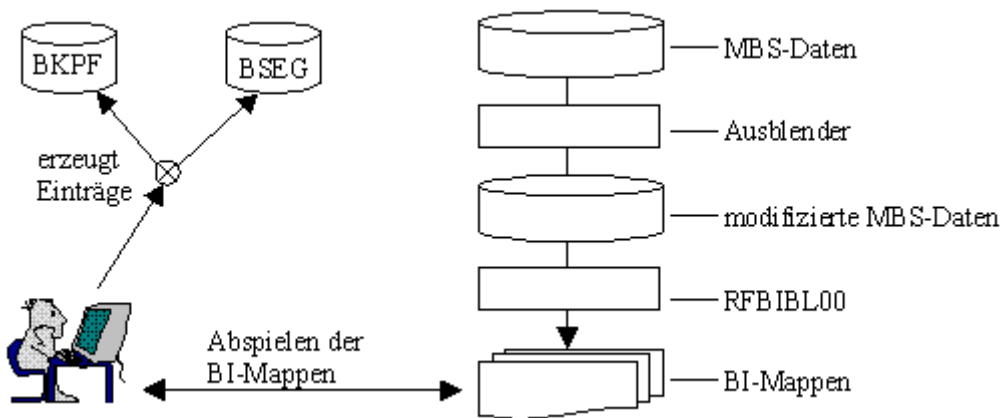
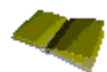


Abbildung 11 - Schematischer Ablauf der Anwendungen und Objekte

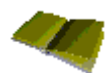


## 6.2 SAP-interne automatische Feldfüllung

Vergleicht man die Sätze, die ursprünglich aus der UNIX-Datei kamen, mit denen, die in den Tabellen BSEG und BKPF generiert wurden, so fällt auf, daß verschiedene Felder u.U. nicht korrekt übernommen wurden. Über mögliche Gründe der Abweichungen wurde bereits gesprochen. Doch bei diesen Fällen bleibt es meistens nicht.

Es kann sein, daß Buchungen auf dem Host nur den Währungsrechnungsbetrag gefüllt haben, in SAP jedoch sowohl den Währungsrechnungsbetrag als auch den DM-Betrag. Das kann passieren, wenn SAP in seinen Modulvorschriften solche Felder mit dem entsprechenden Währungsschlüssel automatisch füllt.

Bei einem Vergleich des Feldes muß ein Prüfprogramm einen Übereinstimmungsfehler ausweisen, obwohl eigentlich keine echte Abweichung vorliegt. Das zeigt zwar, daß das Programm korrekt arbeitet, inhaltlich gesehen liegt jedoch kein manueller Benutzereingriff vor. Da es in der Hauptsache um diese manuellen Eingriffe geht, ist dieses Ausweisen nicht erwünscht.



## 6.3 Revisionssicherheit

Aus Gründen der Buchungsnachweispflicht ist es gesetzlich vorgeschrieben, nachzuweisen, daß Buchungen, die auf dem Host erfolgt sind, 1:1 nach SAP übernommen worden sind. Da ein Prüfprogramm im oben genannten Fall einen Fehler ausweisen würde, obwohl keine manuelle Änderung von statten gegangen ist, bleibt die Frage nach einer aussagekräftigen Lösung.

Allerdings existiert auch eine Meinung, die sich gegen diese Nachweispflicht ausspricht.

Ausgehend von dem Standpunkt, daß letztendlich nur für das aktuelle Buchhaltungssystem die Einträge nachgewiesen werden können müssen, ist das MBS-System nicht von Bedeutung, da das aktuelle System nun einmal SAP ist. Belege müssen im SAP gefunden werden. Dabei ist es egal, ob Daten vom MBS ins SAP korrekt übergeben wurden. Ein Mißbrauch hätte schon auf Ebene des MBS-Systems geschehen können, und dieser wäre ja dann nicht nachweispflichtig. Sollte das Mappenprotokoll Eingriffe beim Abspielen der BI-Mappe ausweisen, steht der Username, also der verantwortliche Buchhalter, dabei. Damit wäre der Nachweispflicht Genüge getan.

Eine Diskussionsrunde, bestehend aus Vertretern der Revision und der Informatik, erörterte Anfang Juni dieses Thema, und kam zu folgendem Ergebnis:

Es sind grundsätzlich die Belege nachzuweisen, die manuell bearbeitet worden sind.

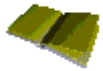
Wird beim Abspielen einer Mappe nicht manuell eingegriffen, so steht normalerweise der Name eines Batch-Users im Protokoll.

Angelegt		Mappenname	User
26.05.1999 14:30:18		ZFS_101LI	R00 B1rwq100
Zeit	Tran	Dynpro	Nachricht
14:30:18			S00300 Mappe ZFS_101LI wird abgepielt durch Benutzer R00 im Modus
14:30:29	FB01		SF5312 Beleg 4000001071 wurde im Buchungskreis 0001 gebucht
14:30:29			S00370 Abspielstatistik
14:30:29			S00363 1 Transaktionen gelesen
14:30:29			S00364 1 Transaktionen verarbeitet
14:30:29			S00365 0 Transaktionen fehlerhaft
14:30:29			S00366 0 Transaktionen gelöscht
14:30:29			S00382 Batch-Input-Verarbeitung beendet

Abbildung 12 - Mappe mit manuellen Eingriff

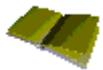
Wird jedoch manuell eingegriffen, egal aus welchem Grund, steht der Username desjenigen im Protokoll, der für diese Änderung verantwortlich zeichnet.

In diesem Fall soll die Person, die manuell eingegriffen hat, Aussage darüber machen, warum sie eingegriffen hat, und was sie geändert hat. Desweiteren muß die BI-Schnittstellendatei mit den Host-Buchungen täglich archiviert werden. Wenn dies gewährleistet ist, können die Mappen mit den Protokollen gelöscht werden.



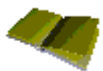
## 6.4 Datenmenge der Protokolle

Bei jeder Buchung erzeugt das SAP-System einen Protokollsatz, den das Verbuchungsprogramm einliest und erst auf Grund dessen ein „commit work“ absetzt, was den Datenbankserver die eigentliche Buchung durchführen läßt. Normalerweise wird nach erfolgreichem Ändern der Protokollsatz aus der Protokollsatzdatei gelöscht. Dies ist jedoch nicht erlaubt, wenn Daten aus einem anderen System per BI importiert werden. In einem Unternehmen wie der BAYER AG fallen selbst in einem eingeschränkten Bereich noch so viele Buchungen an, daß sich eine Protokolldatei von mehreren GIGABYTE bildet. Da das Belegvolumen stetig steigt, muß eine Lösung gefunden werden, um gesetzeskonform die Protokolle löschen zu dürfen. Eine denkbare Möglichkeit wäre, die gesamte Protokolldatei optisch zu archivieren. Dies scheitert jedoch zum einen an der technischen Problematik, sowie an der Tatsache, daß zu jeder der Buchungen sofort auf den Beleg zugegriffen werden können muß.



## 6.5 Lösungsansätze

Der beschriebene Zustand wurde nach einer IST-Analyse im Februar 1999 vorgefunden. Zu diesem Zeitpunkt fing ich mit meiner Arbeit an, die daraus bestand, einen ABAP/4-Abgleichreport zu schreiben. Das Ergebnis sollte ein Report sein, der jede Buchung 1:1 mit der UNIX-Datei aus MBS vergleicht, und ausweist, welche Belege korrekt gebucht wurden, bei welchen Belegen Abweichungen auftraten (warum auch immer), und welche Belege, die im MBS gebucht worden sind, nicht in den SAP-Buchungs-Tabellen wiederzufinden sind.



# Archivierung

## Einleitung

## Dokumentenarchivierung

## Datenarchivierung

### Allgemeines

### Technik der Datenarchivierung

#### Ablage der Daten

Ablage im Filesystem (mit Sicherung  
und/oder Spiegelung)

Ablage in einem Hierarchical Storage  
Management (HSM)

Ablage in einem Archivsystem über  
ArchivLink  
Common Store

## SAP ArchivLink

## FI-Belege

## Index-, Konto- und Beleglaufzeiten

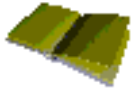
## Zusammenfassung

---

Im Zuge des Medienwechsels von Papier/Microfilm zu optischen Archivsystemen, z.B. Jukeboxen, erfährt die optische Archivierung eine immer größere Bedeutung. Durch rechtliche Vorschriften und/oder interne Normen sind Unternehmen verpflichtet, Dokumente eine bestimmte Zeit lang aufzubewahren und damit zu archivieren.

Traditionellerweise werden Papierarchive oder Microfiches dazu verwendet, die jedoch mit hohen Kosten verbunden sind, sei es für das Kopieren selbst oder die Räumlichkeiten zur Lagerung. Optische Medien dagegen sind kostensparend und bedeuten einen Schritt in Richtung optimalere Ressourcenaufteilung.

In diesem Kapitel soll anfangs auf die grundlegenden Aufgaben und Abläufe der Archivierung unter SAP R/3 eingegangen werden. Nach dem Vorstellen der Grundlagen wird kurz das Archivieren von FI-Daten erläutert.



## 7.1 Einleitung

Ein stark wachsendes Datenvolumen kann bei heutigen R/3-Systemen zu Leistungsengpässen führen, die sich auf der Anwenderseite in einer schlechten Performance und auf der Administrationsseite in einem erhöhten Ressourcenaufwand niederschlagen.

Deshalb sollen Daten, auf die nur noch selten zugegriffen und die nicht mehr geändert werden, aus der Datenbank entfernt werden. Da diese Daten jedoch u.U. nicht einfach gelöscht werden dürfen, da sie aufbewahrungspflichtig sind, kommt hierbei die Datenarchivierung zum Einsatz. Diese ermöglicht das Auslagern von Anwendungsdaten aus der R/3-Datenbank, wobei jedoch aus dem R/3 System weiterhin lesend auf diese Daten zugegriffen werden kann.

Für das Archivieren von Daten in der Finanzbuchhaltung liegen systembedingte und gesetzliche Gründe vor:

- die durch das kontinuierliche Wachstum der Bewegungsdaten verursachten Speicherplatz- und Laufzeitprobleme werden verbessert
- die Stammdaten werden handhabbarer und können auf einem aktuellen Stand gehalten werden
- aus gesetzlichen Gründen ist es erforderlich, Daten in Archiven zu halten, um sie hierdurch zu einem späteren Zeitpunkt wieder zugreifbar zu machen

Daten, die im Online-System nicht mehr benötigt werden, können mit speziellen Standardfunktionen archiviert werden. Die Daten werden dabei in Betriebssystem-Dateien (sog. Archivdateien) gespeichert und i.a. aus dem Online-System gelöscht.

Aus gesetzlichen und betriebsinternen Gründen ist es erforderlich, die in den Archivdateien ausgelagerten Daten unter bestimmten Umständen zu einem späteren Zeitpunkt wieder im Online-System verfügbar zu haben. Dies wird durch das Rückladen erreicht.

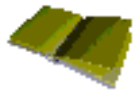
Das SAP R/3-System unterstützt eine Vielzahl von Geschäftsprozessen, die eine Bearbeitung von Dokumenten in Papierform, aber auch in elektronischer Form, beinhalten. Die Dokumente sind entweder Auslöser für die verschiedensten Bearbeitungsschritte in der Finanzbuchhaltung, dem Vertrieb, der Personalverwaltung und vielen anderen Bereichen, oder sie entstehen als Dokumente von Bearbeitungsprozessen in diesen betriebswirtschaftlichen Bereichen.



Wenn sich heute im SAP-Umfeld mit dem Begriff Archivierung beschäftigt wird, trifft man dort die verschiedensten Begrifflichkeiten zum Thema Archivieren. Man begegnet den Begriffen ArchiveLink, Archivierung, Belegarchivierung, Ablage und Reorganisation und stellt schnell fest, daß eine gewisse Unsicherheit im Umgang mit den Begriffen besteht.

Im Folgenden sind mit diesen Begriffen zwei, in ihrer betriebswirtschaftlichen Bedeutung völlig unterschiedliche, Verfahren der Archivierung beschrieben. Die SAP unterscheidet zwischen Dokumentenarchivierung und Datenarchivierung. Im folgenden werden beide Begriffe erläutert und gegeneinander abgegrenzt.

Dieses Kapitel orientiert sich am SAP R/3 Releasestand 4.0B. Die getroffenen Aussagen gelten jedoch größtenteils auch für die Releasestände 3.x. Im Laufe des Release 4.5 wird die SAP eine Änderung der Begriffe vornehmen. Darüber hin-aus soll die ArchiveLink Schnittstelle durch ein BAPI\* Interface ersetzt werden.



## 7.2 Dokumentenarchivierung

Dokumentenarchivierung wird auch mit den Begriffen ArchivLink, Archivierung, Belegarchi-vierung und Ablage bezeichnet.

### **Definition:**

**Dokumentenarchivierung ist das elektronische Ablegen von Originalbelegen.**

In der Dokumentenarchivierung lassen sich folgende Funktionalitäten unterscheiden:

**Eingehende Dokumente** z.B. Rechnungen; das Dokument wird eingescannt und über Archivierung dem R/3 Beleg zugeordnet

**Ausgehende Dokumente** z.B. Bestellungen und Lieferscheine; das Dokument wird im Rahmen der Ausgabe aus dem R/3 System als „Carbon Copy“ archiviert und dem zugehörigen R/3 Beleg zugeordnet

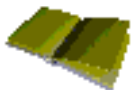
**Drucklisten** z.B. Chargenverwendungsnachweise, Bilanzbewertungen und Kostenstellenberichte; diese Listen werden als Ergebnis von Auswertungen im R/3 archiviert. Dies ist der der Diplomarbeit zugrundeliegende Fall.

Dokumentenarchivierung läuft ausschließlich über die ArchiveLink-Schnittstelle. Ein Archivsystem ist deshalb für den produktiven Einsatz zwingend notwendig. Bei der BAYER AG gibt es folgende Konzernvorgaben bzgl. Speicherart und Dokumentenarchivierung:

- Da die Dokumente nicht veränderbar abgelegt werden müssen, ist eine optische Speicherung der archivierten Daten notwendig.
- Als Archivierungssystem im Bereich Dokumentenarchivierung ist Filenet der Standard im Bayer Konzern.

Die gleichen Archivierungsanforderungen wie für die Dokumentenarchivierung gelten auch für die Ablage von Systeminformationen. Die Archivierung dieser Informationen ist im R/3 Standard nicht vorgesehen.

Unter Systeminformationen (nicht Standard) versteht man z.B. Transportaufträge, Jobproto-kolle, **Batch Input Mappen Protokolle**, Syslogeinträge, Verbuchungsabbrüche und ähnliches.



## 7.3 Datenarchivierung

### 7.3.1 Allgemeines

Datenarchivierung wird von der SAP auch mit den Begriffen Archivierung und Ablage bezeichnet. Als historischer Begriff aus der R/2 Welt wird ab und zu auch

Reorganisation verwendet.

### **Definition:**

**Datenarchivierung ist das Auslagern von Datensätzen aus Datenbanktabellen, mit dem Ziel, die im System vorhandenen Datenbanktabellen mengenmäßig, d.h. die Anzahl der in ihnen gespeicherten Datensätze, zu verkleinern.**

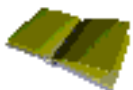
Ein zentrales Element in der Datenarchivierung ist das Archivierungsobjekt. Es definiert die kleinste Einheit, die als Ganzes archiviert und in der Datenbank gelöscht werden kann. Durch ein Archivierungsobjekt wird ein Anwendungsobjekt typisiert, um es für die Archivierung nutzbar zu machen. Es beschreibt, wie und auf welche Datenbankobjekte zugegriffen werden muß, um ein betriebswirtschaftliches Objekt vollständig zu archivieren.

Ein Archivierungsobjekt setzt sich im wesentlichen aus drei Komponenten zusammen:

- |                                 |  |
|---------------------------------|--|
| <b>Datendeklarationsteil</b>    | Hier werden alle relevanten Datenbankobjekte beschrieben, die ein Anwendungsobjekt charakterisieren. |
| <b>Archivierungsprogramme</b>   | Dazu gehören z.B. Schreibprogramm, Löschmodprogramm, Anzeigeprogramm, Rückladeprogramm               |
| <b>Customizingeinstellungen</b> | Damit werden archivierungsobjektspezifische Parameter zu einem Archivierungslauf eingestellt.        |

Die Funktionen für die Archivierung und den späteren Zugriff auf die archivierten Daten, werden durch das Archive Development Kit (ADK) zur Verfügung gestellt. Das ADK kann die archivierten Daten sowohl im Filesystem (File, HSM, Band) als auch über die ArchiveLink-Schnittstelle in einem Archivsystem ablegen und verwalten.

Wenn ein Archivsystem für die Ablage der Archivierungsobjekte verwendet wird, so ist hoher Durchsatz und ein großes, kostengünstiges Speichervolumen entscheidend. Wegen der speziellen Ablagetechnik in der Datenarchivierung ist eine Recherche über die abgelegten Archivierungsobjekte nur mit SAP Mitteln sinnvoll.



## 7.3.2 Technik der Datenarchivierung

### 7.3.2.1 Ablage der Daten

Bei der R/3-Datenarchivierung werden die archivierten Belege aus der Datenbank komprimiert und in Dateien geschrieben. Diese Dateien können mit den Mitteln des ADK gelesen werden. Es stellt sich zunächst die Frage, wie die Dateien am besten verwaltet werden können. Folgende Speicherkonzepte sind möglich:

- Ablage im Filesystem (mit Sicherung und/oder Spiegelung)
- Ablage in einem Hierarchical Storage Management System (HSM)
- Ablage in einem Archivsystem über ArchiveLink

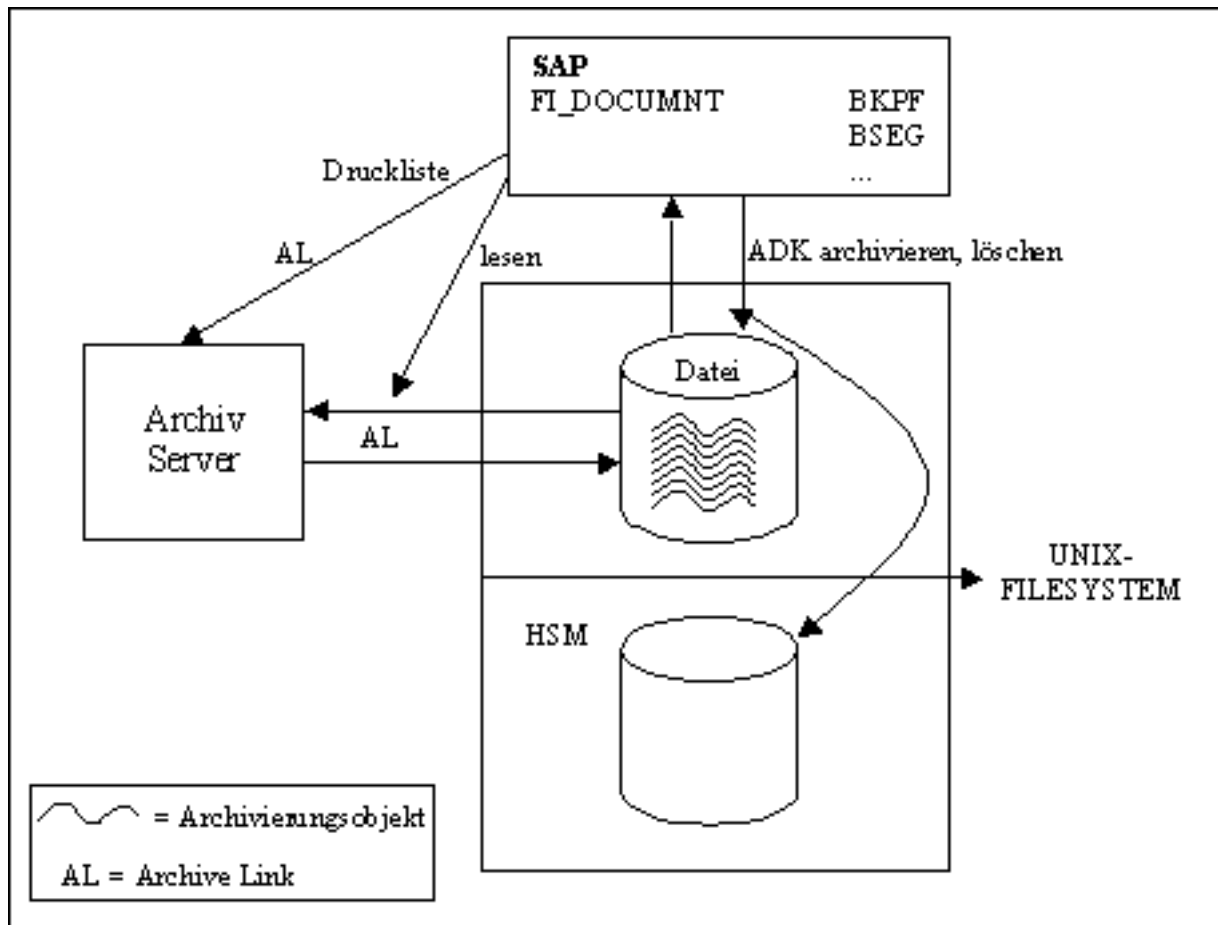
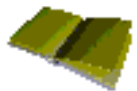


Abbildung 13 - Zusammenspiel ADK und ArchiveLink [7]



#### 7.3.2.1.1 Ablage im Filesystem (mit Sicherung und/oder Spiegelung)

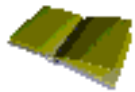
In diesem Fall bleiben die vom Archivierungslauf geschriebenen Dateien im Filesystem. Aus R/3-Sicht ist dies der einfachste Fall (und von der Zugriffsgeschwindigkeit aus R/3 heraus auch der Schnellste). Es ist auch keine

ArchiveLink-Anbindung nötig.

Nachteilig an dieser Lösung ist, daß auf längere Sicht große Filesystemkapazitäten benötigt werden. Da der Platz im Filesystem relativ teuer ist und als unsicher gilt, scheidet diese Ablage in der Praxis meist aus. Desweiteren sind die Kosten für dieses Speichermedium im Vergleich zu anderen Alternativen zu hoch.

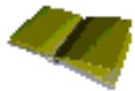
Kurzfristig ist es trotzdem eine Lösung, da die Daten aus der DB herausgelöscht sind, und die DB-Zugriffe dadurch schneller werden.

Eine echte Laufzeitverbesserung erreicht man schließlich durch eine Reorganisation der Oracle-Datenbank auf dem Server. Doch ein solcher Reorg-Lauf würde auf dem Produktivsystem F03 über 24 Stunden dauern, und wird deswegen so gut wie nie gemacht.



#### 7.3.2.1.2 Ablage in einem Hierarchical Storage Management System (HSM)

Eine Alternative zum Filesystem ist die Ablage in einem HSM-System. R/3-seitig unterscheidet sich diese Lösung nicht von der Filesystem-Lösung. Das HSM-System stellt für R/3 ein beliebig großes Filesystem zur Verfügung, wobei das HSM die tatsächliche Ablage der Daten übernimmt. Es ist jedoch zu beachten, daß die Zugriffszeit wesentlich schlechter als bei Zugriff auf das Filesystem sein kann (je nachdem, auf welches Medium das HSM die Daten ausgelagert hat).



#### 7.3.2.1.3 Ablage in einem Archivsystem über ArchiveLink

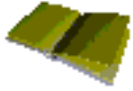
Hierbei werden die Dateien nach erfolgtem Löschlauf über ArchiveLink an ein Archivsystem übergeben und anschließend aus dem Filesystem gelöscht. Im Unterschied zu den beiden erst-geannten Methoden ist hierfür die ArchiveLink-Schnittstelle nötig. Insbesondere muß das verwendete Archivsystem ArchiveLink 3.1 zertifiziert sein. Ein Archivsystem kann je nach Anforderung die Daten optisch oder magnetisch speichern.

Bei späterem Zugriff auf die Dateien wird stets die ArchiveLink-Schnittstelle aktiv. Dies passiert jedoch, ohne das der Benutzer es explizit veranlassen muß.

Nachteil der Ablage auf ein Archivsystem über ArchiveLink ist der größere (Zeit-)Aufwand, der durch die ArchiveLink-Schnittstelle entsteht. Desweiteren muß im R/3-System ein entsprechendes Customizing vorgenommen werden, um die Verbindung zwischen R/3-System und Archivsystem herzustellen.

Andererseits entfällt bei dieser Methode der Verwaltungsaufwand für die Dateien. Außerdem gewährleistet ein Archivsystem größere Datensicherheit als

## Filesysteme.



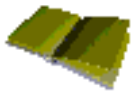
### 7.3.2.1.4 Common Store

IBM Common Store ist ideal geeignet, alte, nicht mehr dringende Daten aus produktiven R/3 Datenbanken in ein externes Archiv für gelegentliches Zurüchluden und Langzeitspeicherung abzulegen.

Desweiteren können wichtige Geschäftsdokumente in elektronischer Form gespeichert, verwaltet und im Firmennetz publiziert werden. Von dort können sie via SAPGUI oder einem Webbrowser angesehen werden.

In Verbindung mit den führenden Archivsystemen kombiniert IBM Common Store die Funktionen eines zertifizierten SAP Archives mit denen eines umfassenden Unternehmensdokumentenmanagementsystem. Das erlaubt R/3-Anwendern, sowohl SAP-Dokumente als auch SAP-fremde Dokumente in einem gemeinsamen Archiv zu verwalten.

Diese Lösung wurde seit einiger Zeit favorisiert und kommt definitiv ab August 1999 bei der BAYER AG zum Einsatz. [11]

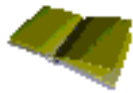


## 7.4 SAP ArchivLink

SAP ArchiveLink ist eine in die Basiskomponente des SAP R/3-Systems integrierte Kommunikationsschnittstelle zwischen den R/3-Anwendungskomponenten und externen Komponenten.

Vereinfacht dargestellt wird ArchiveLink von den R/3 Anwendungskomponenten benutzt, um die Daten an ein Archivsystem zu übergeben und in der Folge auf diese Daten wieder zuzugreifen.

Die Verwirrung in der Begrifflichkeit entsteht nun dadurch, daß die ArchiveLink Schnittstelle zum einen von den R/3 Anwendungsprogrammen (FI, SD, MM,...) zur Dokumentenarchivierung verwendet wird, zum anderen aber auch über die R/3 Basisanwendung ADK eine Anbindung an die ArchiveLink-Schnittstelle realisiert, um die in der Datenarchivierung zu archivierenden Daten an externe Archivsysteme übergeben zu können. ArchivLink unterscheidet die unterschiedliche Nutzung gleicher Funktionalität verschiedener Anwendungen durch Verwendung anderer Dokumenttypen.



## 7.5 FI-Belege

Dieser Abschnitt ist auf die Besonderheiten des Moduls FI ausgerichtet.

Zu dem Modul FI gibt es verschiedene Archivierungsobjekte, so unter anderem FI\_DOCUMNT, FI\_MONTHLY, FI\_SCHECK, FI\_ACCRECV und FI\_BANKS.

In dieser Arbeit wird nur auf das Objekt FI\_DOCUMNT eingegangen.

Finanzbuchhaltungsbelege werden beispielsweise mit diesem Archivierungsobjekt archiviert, gelöscht und zurückge-laden. Es besteht u.a. aus folgenden Segmenten:

Segment	Beschreibung
BKPF	Belegkopf Buchhaltung
BSEG	Belegsegment Buchhaltung
BSET	Belegsegment Steuerdaten
BVOR	Buchungskreisübergreifender Vorgang

Tabelle 1 - Segmente des Objektes FI\_DOCUMNT

Bei der Belegarchivierung werden die archivierbaren Belege inklusive ihrer Änderungsbelege und Langtexte in eine oder mehrere Archivdateien geschrieben. Das Archivierungsprogramm prüft die Archivierbarkeit auf Belegkopf- und Belegpositionsebene. Wenn die Bedingungen für eine Archivierung erfüllt sind, können die Belege archiviert werden; zu diesen Kriterien gehört die Belegartenlaufzeit, die Anzahl Tage im System, die Prüfung auf offene Posten und Konten-artenlaufzeit.

Zu einem guten Archivkonzept gehört, sich einzelne Buchhaltungsbelege aus dem Archiv anzeigen lassen zu können. Dies ist nur dann möglich, wenn sich der Sekundärindex\* noch im System befindet.

Um eine Vorstellung von den Datenmengen zu bekommen, die archiviert werden sollen, werden hier einige Daten aus dem FI-Bereich aufgeführt:

**IST-Zahlen** zum 10.04.1999:

*FI*-Belege:

F03: 1,8 Mio BKPF + 6 Mio BSEG

IH3: 0,25Mio BKPF + 1,1 Mio BSEG

## ***Merkmale für Archivierungsobjekte:***

### **Allgemeine Angaben**

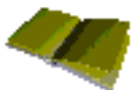
Modul	FI
System	F03 (Produktiv-FI-System)
Mandant	530
Archivierungsobjekt	FI_DOCUMNT
Klasse	Bewegungsdaten

### **Volumen und Laufzeiten**

Anzahl pro Jahr	10 Mio. / Jahr (F03)
Ø Größe eines Objektes	Ca. 2 KB
Gesetzliche Aufbewahrungsfrist	10 Jahre + lfd. Geschäftsjahr
Mindestlaufzeit im System	4 Monate

### **Zugriff auf archivierte Objekte**

Online-Zugriff auf einzelne Objekte	Ja
Akzeptable Zugriffszeit	30 Sek. Lfd. Geschäftsjahr
Zugriff im Batch	Ja, vorläufig nur Standardreports auf Archiv-Belege
Welche anderen Module sind betroffen	MM, CO, SD
Zurückladen erforderlich	Nur im Fehlerfall



## **7.6 Index-, Konto- und Beleglaufzeiten**

Neben der Kontoartenlaufzeit, welche den Zeitpunkt der Archivierung eines



Beleg festlegt, kann auch eine Indexlaufzeit für die sog. Sekundärindizes der Sach-, Debitoren und Kreditorenkonten festgelegt werden. In diesen werden Informationen zum Beleg parallel zu den eigentlichen Belegdaten gehalten. Mittels der Indexlaufzeit kann festgelegt werden, ob und wie lange ein Sekundärindex zu einem Beleg bezogen auf ein Buchungsdatum und Archivierungstichtag über die Archivierung hinaus im System gehalten wird. Dies bedeutet, daß die Sekundärindizes nach dem Archivieren nicht sofort gelöscht werden, sondern erst nach einem Zeitraum, der von der Indexlaufzeit festgelegt wird.

Die folgende Abbildung zeigt exemplarisch das Zusammenspiel der Laufzeiten.

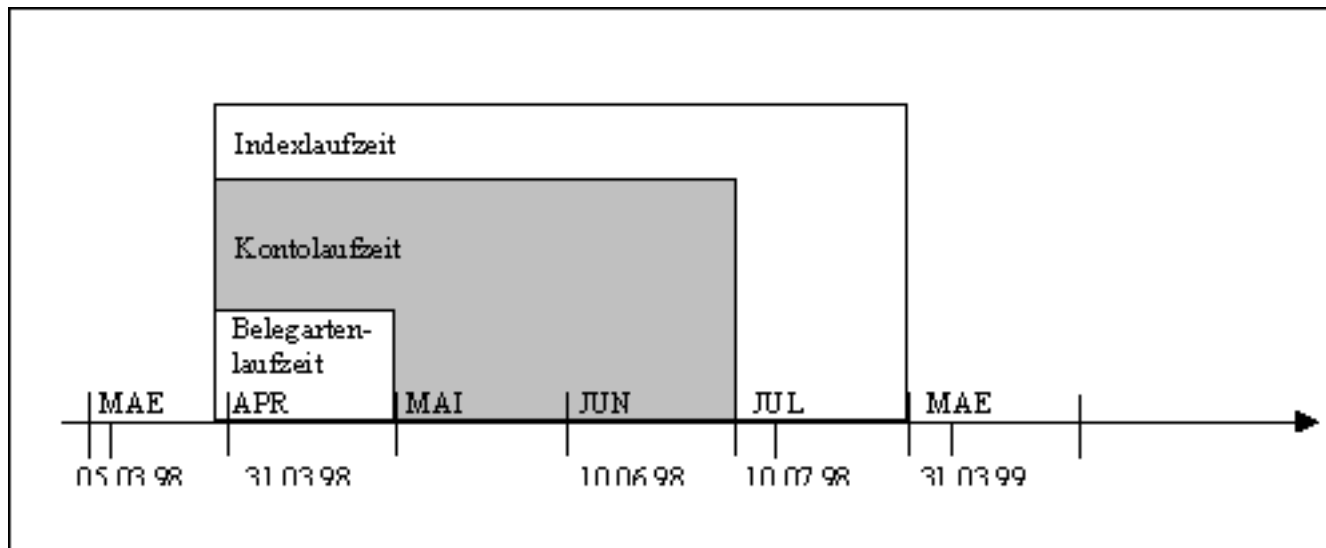


Abbildung 14 – Laufzeiten [9]

Zu jedem Archivierungsprogramm (Archiv schreiben, löschen, zurückladen) gibt es ein Standardprotokoll und ein Detailprotokoll.

Das Standardprotokoll enthält statistische Informationen zur Anzahl der archivierten Datenobjekte.

Im Detailprotokoll wird jedes einzelne Objekt (z.B. jeder am Archivierungslauf beteiligte Beleg) aufgelistet mit der Information, was mit ihm gemacht wurde. Beim Archivierungsprogramm wird beispielsweise für jeden einzelnen Beleg die Information mitgegeben, ob er archiviert wurde oder nicht. Falls der Beleg nicht archiviert werden konnte, wird ein Grund dafür angegeben.

Über die Aktion Zurückladen können archivierte Daten aus den Archivdaten in die Datenbank zurückgeladen werden.

Archivdateien können komplett oder teilweise zurückgeladen werden. Die Archivdatei wird dabei in jedem Fall als zurückgeladen gekennzeichnet. Nicht zurückgeladene Objekte werden in eine neue Archivdatei geschrieben.

Genauso wie Archive können archivierte FI-Belege in das System zurückgeladen werden. Über entsprechende Intervalle kann die Menge der Daten eingeschränkt werden. Der Beleg trägt nach dem Rückladen keine besondere Kennzeichnung und ist damit in seinem Originalzustand.

Der Archivierungs-, Lösch- oder Rückladevorgang wird über Jobs angestoßen. Dabei wird immer der gleiche Job gestartet:

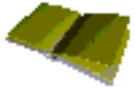
*ARCHIVE\_<OBJEKT>\_SUB\_<DATUM>.*

Dieser dient als Starter für einen der folgenden Jobs:

*ARCHIVE\_<OBJEKT>\_WRI\_<DATUM>.*

*ARCHIVE\_<OBJEKT>\_DEL\_<DATUM>.*

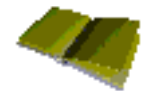
*ARCHIVE\_<OBJEKT>\_REL\_<DATUM>.*



## 7.7 Zusammenfassung

SAP unterscheidet zwei Arten von Archivierung, die Daten- und die Dokumentenarchivierung. Die Archivierungsarten sind funktional und in ihrer betriebswirtschaftlichen Ausrichtung völlig unterschiedlich.

Die SAP ArchivLink Schnittstelle wird also sowohl für die Dokumentenarchivierung als auch für die Datenarchivierung genutzt. Die Ablage im Archivsystem wird von der jeweiligen Anwendung in Zusammenarbeit mit dem ArchiveLink Interface durchgeführt.



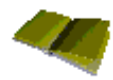
# Reportlogik

## Logik

## Archivierung

### 8. Reportlogik

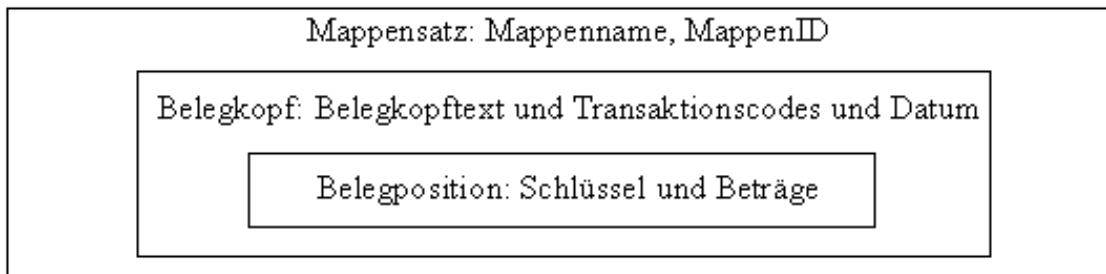
Meine Diplomarbeit richtete sich im ersten Teil darauf aus, einen Report zu entwickeln, der die MBS-Buchungen mit den Buchungen im SAP-System abgleicht und das Resultat ausweist.  
In einem zweiten Schritt soll ein Konzept zur Archivierung der aus dem Report resultierenden Liste erstellt werden.  
Die Mappen, die zum Buchen dieser FI-Belege abgespielt werden, und die dazugehörigen Protokolle sollen in einem dritten Schritt gelöscht und die BI-Protokolldatei reorganisiert werden.



#### 8.1 Logik

Der Ansatzpunkt dieser Arbeit ist der Moment, wo die MBS-Daten über die Schnittstelle zu SAP neu verbucht werden. Die Daten der Eingabe-Datei müssen gemäß den Schnittstellenanforderungen bereits in einer speziellen Strukturform vorliegen.

Damit ergibt sich folgendes Bild für den Satzaufbau der BI-Schnittstellendatei (s. Abb. 18):



Der beschriebene Dateiaufbau soll anhand einer Beispieleingabedatei visualisiert werden:

Struktur		Inhalt (erste 100 Bytes)
BGR00	neue Mappe	0ZFS_101AB 540WF-BATCH-S 1/
BBKPF	Kopfdaten	1FB0115041999SB00011604199904USD //
BBSEG	Folgedaten	2BBSEG 40/ / 0000000000000,00000000000001,14/
BGR00	neue Mappe	0ZFS_101BS 540WF-BATCH-S 1/
BBKPF	Kopfdaten	1FB0114041999SB00011604199904EUR //
BBSEG	Folgedaten	2BBSEG 50/ / 0000000029442,690000000057584,90/
BBSEG	Folgedaten	2BBSEG 40/ / 0000000034153,520000000066798,48/
BBTAX	Folgedaten	2BBTAX 0000000004710,83BQ50 0000000009213,58
BBKPF	Kopfdaten	1FB0114041999SB00011604199904EUR //
BBSEG	Folgedaten	2BBSEG 50/ / 0000000028296,110000000055342,38/
BBSEG	Folgedaten	2BBSEG 40/ / 0000000032823,490000000064197,17/
BBTAX	Folgedaten	2BBTAX 0000000004527,38BQ50 0000000008854,79
BGR00	neue Mappe	0ZFS_101EA 540WF-BATCH-S 1/
BBKPF	Kopfdaten	1FB0114041999SB00011604199904EUR //
BBSEG	Folgedaten	2BBSEG 40/ / 0000000039440,000000000077137,94/
BBSEG	Folgedaten	2BBSEG 50/ / 0000000039440,000000000077137,94/
BBKPF	Kopfdaten	1FB0115041999SB00011604199904FRF //
BBSEG	Folgedaten	2BBSEG 40/ / 0000001628400,000000000485530,85/
BBSEG	Folgedaten	2BBSEG 50/ / 0000001628400,000000000485530,85/
BGR00	neue Mappe	0ZFS_101EN 540WF-BATCH-S 1/
BBKPF	Kopfdaten	1FB0114041999SB00011604199904PTE //
BBSEG	Folgedaten	2BBSEG 50/ / 0000000018468 0000000000180,17/
BBSEG	Folgedaten	2BBSEG 50/ / 0000000004358 0000000000042,52/
BBKPF	Kopfdaten	1FB0114041999SB00011604199904JPY //
BBSEG	Folgedaten	2BBSEG 50/ / 00000000230920 00000000003513,34/
BBSEG	Folgedaten	2BBSEG 50/ / 0000000011734 0000000000178,53/
BBKPF	Kopfdaten	1FB0114041999SB00011604199904USD //
BBSEG	Folgedaten	2BBSEG 50/ / 0000000000097,42000000000176,63/

Abbildung 18 - Inhalt einer BI-Schnittstellendatei

Ein Satz mit der Satzart (im folgenden mit SA abgekürzt) 0 bezeichnet einen Mappensatz. In seinen Feldern sind Informationen wie der Mappenname und der Mandant gespeichert.

Diesen Namen erhält dann auch die Mappe im SAP-System, wenn sie erstellt wird.

Zu einem Mappenkopfsatz existiert auch mindestens ein Belegkopf, der die SA 1 repräsentiert.

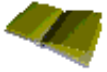
Er stellt das Anlegen eines neuen Beleges an, der im SAP-System eine separate Belegnummer erhält. In der Regel sind das Buchungen, die über die Transaktion <FB01> angelegt werden. In dem Satz mit der SA 1 sind Informationen wie der Transaktionscode, das Beleg- und Buchungsdatum, der Buchungskreis, die Belegwährung, der User und ganz entscheidend für das Wiederfinden des Beleges der Belegkopftext enthalten. Dieser sollte eindeutig im SAP-System sein, denn nur mit seiner Hilfe ist ein MBS-Beleg im SAP-System wiederzufinden. Dort ist dieser Kopf nicht mehr mit einer Mappe verbunden, sondern steht für sich in einer Tabelle. Um jedoch die dazugehörigen Belegpositionen zu finden, muß die Belegnummer bekannt sein. Da diese im SAP-System neu vergeben wird, ist in der UNIX-Datei eine SA 2 zu finden, die die Belegpositionen zu dem Satz der SA 1 darstellt. Diese Position kann eine normale Belegzeile oder eine Steuerbuchung sein. Sie unterscheiden sich durch den Zeilenidentifizier, der BBSEG oder BBTAX heißt.

In der SA 2 finden sich alle Positionsinformationen, die benötigt werden, um ein Positionsdynpro zu füllen und zu buchen. Die Inhalte dieser Dynpros sind beispielsweise Betrag, Buchungsschlüssel, Konten und vieles andere.

Aus dieser Datei werden Mappen erstellt, und diese Mappen werden abgespielt. Daraus resultieren Buchungen, also Einträge in die Buchungstabellen. Nun soll verglichen werden, ob die Felder so gebucht wurden, wie sie in der Datei stehen. Dies geschieht mit dem im Rahmen der Diplomarbeit entwickelten Report:

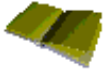
Diese Datei soll nun gelesen, die Satzarten zusammengeführt und als kompletter Beleg im SAP-System wiedergefunden und verglichen werden.

Der Aufbau der SA ist durch die Schnittstellendefinition in SAP vorgegeben. Es existieren spe-zielle Strukturen, die als Grundlage für das Einlesen von BI-Daten vorgegeben sind.



## 8.2 Archivierung

Im Kapitel 7 wurde bereits über die Theorie der Archivierung und der möglichen Archivierungskonzepte gesprochen. Zu Beginn war das Ziel dieser Arbeit, ein Verfahren zur Archivierung der Reportliste zu entwickeln, die BI-Protokollsatzdatei zu reorganisieren, und in den vorhandenen Tagesablauf als Job einzubinden. Das diesbezügliche Problem, das während meiner Diplomarbeit auftrat, war, daß bis dato noch keine echte Archivierungshardware für den Bereich vorhanden war. Aus diesem Grund sind die Themen Archivierung und Reorganisation lediglich theoretisch anzusehen. Dieser Punkt wird voraussichtlich erst ab September realisierbar sein.



## Entwicklungsumgebung

### Korrektur- und Transportwesen (KTW)

#### Systemlandschaft

#### Programmierungsumgebung

##### Allgemeines

##### ABAP/4

##### Tabellen

##### Funktionsbausteine

#### Entwicklung

#### Testumgebung

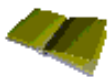
##### Direktes Buchen

##### Abspielen von Batch-Input-Mappen

---

Im Gegensatz zu einer „gewöhnlichen“ Programmiersprache wie z.B. VBASIC oder C/C++ bietet SAP R/3 einen Rahmen, der sich von der Komplexität her vom Standard abhebt.

Zu Beginn dieses Kapitels möchte ich auf das Korrektur- und Transportwesen eingehen, da es eine wesentliche Komponente im SAP R/3 darstellt.



## **9.1 Korrektur- und Transportwesen (KTW)**

Aufgabe einer Entwicklungsumgebung wie der des SAP R/3-Systems ist sowohl die geordnete Erstellung und Modifikation von Objekten (Objekte können Programme, Reports, Dynpros u.ä. sein) innerhalb einer Datenbank als auch der Transport und die Verteilung solcher Objekte über Datenbankgrenzen hinweg.

Das KTW gliedert sich entsprechend dieser Einteilung in zwei Komponenten:

- das Korrektursystem, dessen Aufgaben ab Version 3.0 der Workbench Organizer übernimmt, überwacht die Entwicklungstätigkeit innerhalb eines R/3-Systems;
- das Transportsystem, das für die Verteilung und den Transport zwischen verschiedenen Systemen zuständig ist.

Das Korrektursystem registriert und dokumentiert Änderungen an Objekten der Entwicklungsumgebung. Versucht man, ein Objekt mit dem entsprechenden Werkzeug zu bearbeiten, so wird automatisch das

Korrektursystem aktiviert. Nur wenn der Anwender einen Änderungsauftrag angelegt hat, darf er ein Objekt modifizieren. Dadurch wird das Objekt dem Änderungsauftrag zugeordnet. Durch das Erfassen der Objekte in Aufträgen werden sämtliche Änderungen der ABAP/4 Development Workbench registriert und können somit leicht nachvollzogen werden.

Mit der Verknüpfung von Objekt und Änderungsauftrag schützt das Korrektursystem dieses Objekt vor parallelen oder unkoordinierten Änderungen anderer Anwender.

Nachdem alle Entwicklungsaufgaben abgeschlossen sind, müssen die geänderten Objekte in verbundene R/3-Systeme transportiert werden. Dies heißt in der Regel: die getesteten Programme und Reports müssen vom Test- und Entwicklungssystem in das Produktivsystem transportiert werden.

Beim Transport von Objekten besteht die Möglichkeit:

- Objekte im Zielsystem zu überschreiben
- Objekte im Zielsystem automatisch zu löschen oder
- Objekte ins Zielsystem einzufügen, ohne bereits bestehende Objekte zu überschreiben

Nach dem Abschluß aller Entwicklungsaufgaben muß der Änderungsauftrag freigegeben werden. Das Korrektursystem reicht alle in dem Auftrag verzeichneten Objekte an das Transportsystem weiter. Die in den Korrekturaufträgen registrierten Objekte werden dadurch in einen Transportauftrag übernommen.

Anschließend muß der Transportauftrag freigegeben werden (s. Abb. 19), wodurch alle im Transportauftrag registrierten Objekte exportiert werden. Bei diesem Vorgang werden alle Objekte aus der Quelldatenbank gelesen und in eine Transportdatei auf Betriebssystemebene geschrieben. Zusätzlich werden entsprechende Einträge in die Protokolldateien geschrieben. Anhand dieser Transportprotokolle und der Dokumentation der Änderungsaufträge ist jederzeit rekonstruierbar, welcher Benutzer welche Objekte transportiert hat und zu welchem Zweck dies geschah.

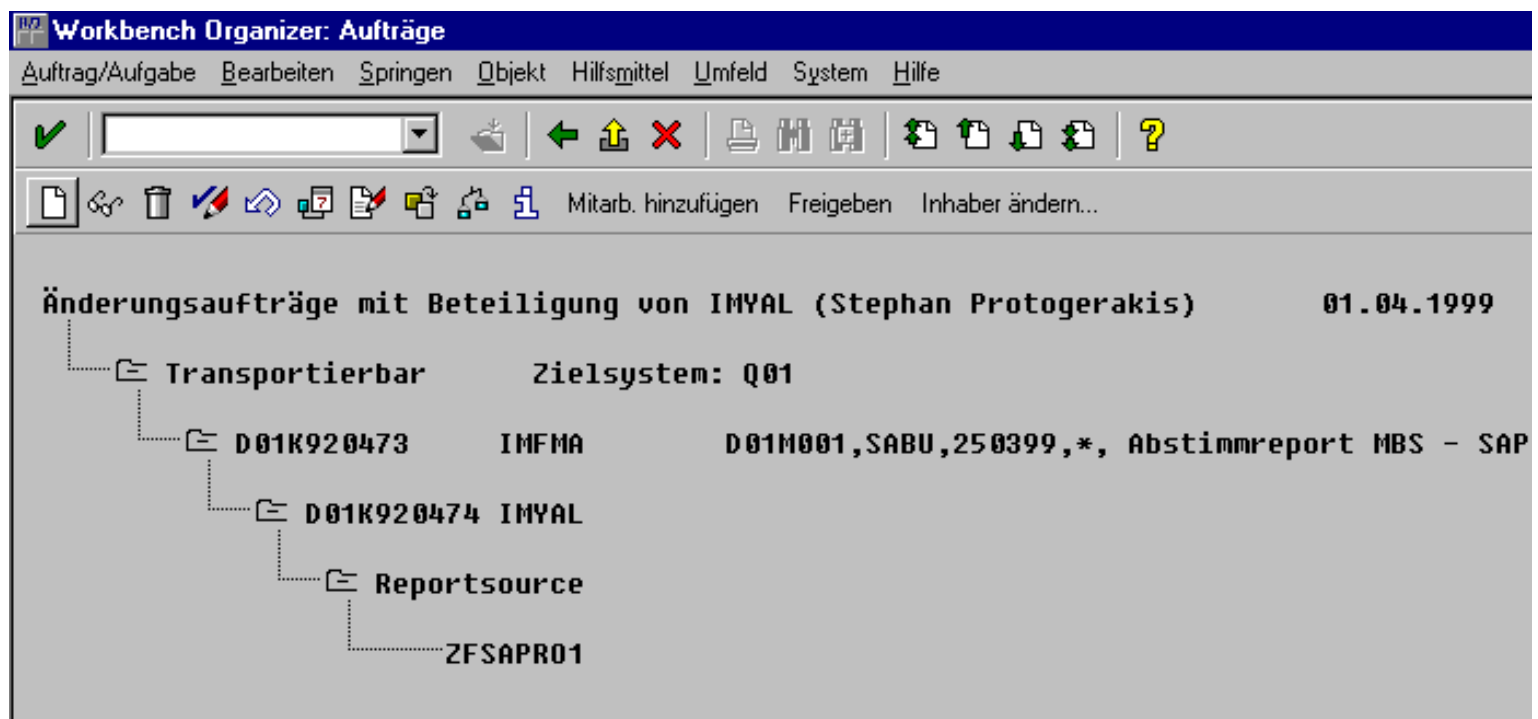
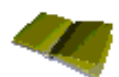


Abbildung 19 - Freigegebener Auftrag



## 9.2 Systemlandschaft

Der wesentlichste Unterschied zu gängigen Programmierumgebungen ist das Drei-Systeme-Konzept von SAP R/3, das der Sicherheit und der Konsistenz laufender Produktivsysteme dient.

Im Gegensatz zu Entwicklungen unter Windows, wo Programme in ein und derselben Umgebung entwickelt, getestet und in Einsatz genommen werden, hat in R/3 jede Stufe der Programmentwicklung seine eigene Umgebung.

In einem von anderen Systemen separaten Entwicklungssystem werden die ersten Ideen bis hin zur fertigen Anwendung entwickelt und realisiert. Mit Hilfe von Testdaten wird das Programm in seiner vollen Ausprägung (also auch vollständige Dynpros mit ihrer integrierten Ablauflogik) auf ihre inhaltliche Richtigkeit getestet.

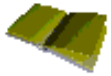
Wenn dieser Test zufriedenstellend verlaufen ist, kann man den Auftrag, unter dem das Programm entwickelt worden ist, in ein Qualitätssicherungssystem (im folgenden als QS-System abgekürzt) transportieren. Dieser Transport wird von dem KTW durchgeführt. Zu diesem Zeitpunkt geht man jedoch bereits davon aus, daß das Programm syntaktisch und weitestgehend auch logisch korrekt ist.

Auf dem QS-System wird der gleiche Test wie auf dem Entwicklungssystem durchgeführt. Hierbei wird jedoch vorrangig getestet, ob das KTW auch alle Teile des Auftrags (Pro-grammcode, Dynpros, ...) sauber transportiert hat. Wenn bei der Übertragung Fehler aufgetreten sind, so würden die selben Fehler bei der Übertragung ins Produktivsystem auftreten, was verheerende Folgen für den laufenden Produktivbetrieb haben könnte.

Ist die Übertragung ins QS-System jedoch erfolgreich verlaufen und alle Features des entwickelten Programmes sind nach wie vor aktiv, kann man auch davon ausgehen, daß eine Übertragung ins Produktivsystem erfolgreich vonstatten geht.

Bei der KTW-Übertragung wird stets ein kompletter Auftrag transportiert, unter anderem auch deshalb, weil nicht wie bei Compilersprachen ein gebundenes Objekt am Ende der Entwicklung steht, sondern die Quellcodes.

ABAP/4 ist eine „Open Source“-Sprache, wo jeder die Quellcodes ansehen und kopieren darf.



## 9.3 Programmierung

### 9.3.1 Allgemeines

Es gibt zwei Grundtypen von R/3-Anwendungen: Online-Transaktionen und Reports.

Transaktionen sind Endbenutzer-Anwendungen, bei denen die Anwendung auf die Dialogeingaben des Benutzers reagiert. Diese Benutzereingaben ziehen sich zumeist über mehrere Dynpros. Man nennt solche Anwendungen auch Dialogprogramme. Dabei werden zumeist auch eigene Dynpros erstellt. Die benutzten Module befinden sich in einem Modulpool.

Bei Reports hingegen werden keine oder nur wenige Benutzeraktionen auf einer Eingabemaske gefordert. Reports dienen daher eher als Analyse-Anwendungen, bei denen aufgrund von Benutzerparametern eine Ergebnisliste dargestellt wird. Solche Anwendungen werden auch als Listreports bezeichnet.

Der im Rahmen der Diplomarbeit entwickelte Abstimmreport ist eine solche Analyse-Anwendung, da er außer einiger weniger Eingabeparameter keinerlei Benutzereingaben benötigt. So sind auch keine weiteren Objekte wie Dynpros, Menüs oder spezielle Screen-Designes in diesem Report notwendig und werden in dieser Arbeit auch nicht näher erläutert.

Das Produkt SAP R/3 verfügt über eine gut ausgeprägte Schnittstelle, um Daten per BI ins System einzuspielen. Zu den schon aufgeführten Problemen gibt es jedoch keine Standardlösung.

Aus diesem Grunde war es notwendig, auf die integrierte Entwicklungsumgebung von SAP R/3 zurückzugreifen und eine eigene Lösung zu entwickeln. Daher folgt nun eine kurze Beschreibung der Umgebung:

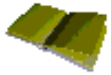
Die Entwicklungsumgebung heißt ABAP/4 (Advanced Business Application Programming/4) und ist eine Programmiersprache der vierten Generation, die SAP zur Erstellung von Anwendungsprogrammen, die nicht im Reportstandard enthalten sind, entwickelt hat.

Die ABAP/4 Development Workbench ist die grafische Entwicklungsumgebung von SAP. In ihr finden sich alle Entwicklungs- und Administrationswerkzeuge, die für die Erstellung einer ABAP/4-Applikation benötigt werden:



- ABAP/4 Dictionary
- ABAP/4-Editor
- Funktionsbibliothek
- Screen Painter
- Menü Painter
- Testwerkzeuge / Debugger

Da diese in der Hauptsache für Dialogprogrammierung vorgesehen sind, soll auf die Komponentenbeschreibung an dieser Stelle verzichtet werden.



### 9.3.2 ABAP/4

Die Programmiersprache ABAP/4 wurde ursprünglich für die Anwendungsentwickler der Firma SAP entworfen. ABAP/4 wurde ständig weiterentwickelt und wird auch von SAP-Kunden im Anwendungsbereich verwendet. Es ist eine Interpreter-Sprache, deren Basis fast vollständig in C, alles darüber hinaus in sich selber geschrieben wurde. Besonders für COBOL-Programmierer ist ABAP/4 sehr leicht zu erlernen, da die Syntax eine bunte Mischung aus COBOL und SQL ist.

Bei firmenspezifischen ABAP/4-Programmierungen, wie im vorliegenden Fall, sollte jedoch genau geprüft werden, ob eine R/3-Standardlösung verwendet werden kann. Der Aufwand für die Pflege und Wartung von ABAP/4-Individuallösungen, beispielsweise bei einem R/3-Versionswechsel, kann leicht unterschätzt werden. Programmentwicklungen unter ABAP/4 sollten gemäß SAP AG nach der Early-Prototyping-Methode durchgeführt werden:

Als erstes wird eine eingeschränkte Programmversion erstellt, der man zu gegebener Zeit noch fehlende Funktionalität hinzufügt.

Es werden immer Zwischenversionen erstellt, die selbst lauffähig sind und bereits vom Endanwender getestet werden können.

Dies ist als Empfehlung, nicht als Vorschrift anzusehen. Es bleibt jedem Entwickler selber überlassen, wie er die Softwareentwicklung handhabt.

Aufgrund ihrer Ablaufsteuerung unterscheidet man folgende Typen von ABAP/4-Programmen:

- Batch-Input-, Dialogprogramme:  
Über Modul-Pools werden Dialogmodule aufgerufen, die in den Dynpros hinterlegt sind, wie Layout, Prüfungen und Ablauflogik. Dies können auch Batch-Input-Programme sein, die Daten aus SAP- oder Fremdsystemen transportieren.
- Verbuchungsprogramme  
Diese werden zur Bearbeitung von Protokollsätzen benötigt
- Reportprogramme  
Über das sogenannte Reporting werden Auswertungen (Listen) aufbereitet
- Include-Programme  
In den Include-Programmen können übereinstimmende Programmteile von mehreren ABAP/4-Programmen genutzt werden
- Kommunikationsprogramme  
Diese erlauben die Steuerung der Rechnerkommunikation

Folgende Abbildung 20 soll dies visualisieren:

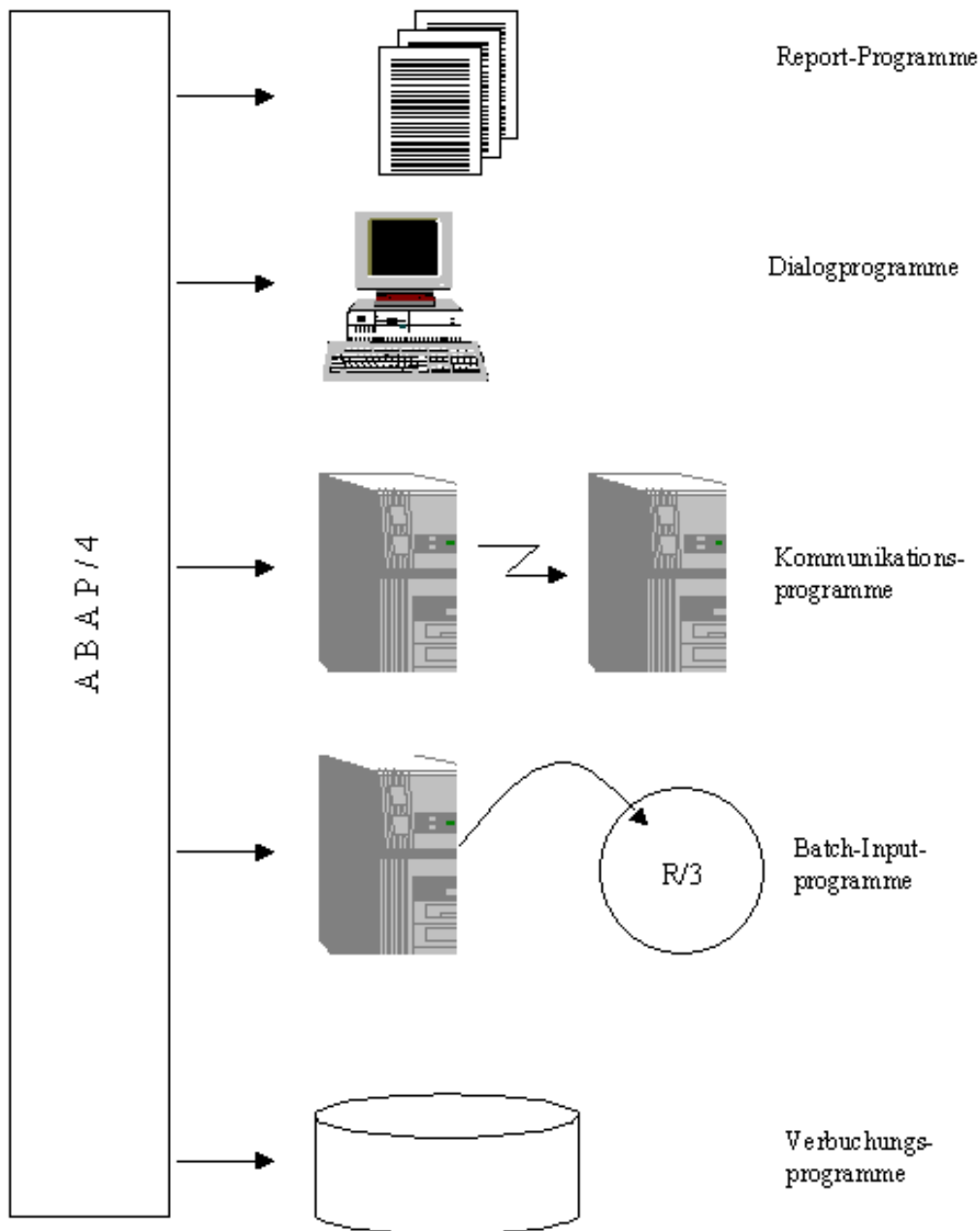
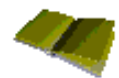


Abbildung 20 - ABAP/4-Programmtypen [2]



### 9.3.3 Tabellen

Jede Tabelle besitzt einen Primärschlüssel. Dieser besteht aus einem oder mehreren Feldern, durch die eine Tabellenzeile eindeutig festgelegt wird.

Die Eigenschaften der Tabellenfelder werden in zwei Stufen festgelegt:

- Datenelement
- Domäne

Die Domäne ist für die Feldbeschreibung die elementarste Grundlage. Jede Domäne erhält einen eindeutigen

Namen und beschreibt Länge und Datentyp, also die technischen Eigenschaften eines Feldes. Sie sind in ihrer Existenz unabhängig von den eigentlichen Tabellen und können somit in eine oder mehrere Tabellen integriert werden. Sobald die Definition einer Domäne geändert wird, ändern sich ebenfalls die Felder, die sich über das ihnen zugewiesene Datenelement auf dieselbe Domäne beziehen. [6]

Die Datenelemente bauen auf Domänen auf, sind genau wie die Domänen tabellenunabhängig und können ebenfalls in eine oder mehrere Tabellen integriert werden. Sie enthalten den Verweis auf eine Domäne, um die technischen Eigenschaften zu definieren, und ergänzen diese mit Angaben zur logischen Bedeutung des Datenelements aus Sicht des Anwenders. Ein Datenelement beschreibt also den betriebswirtschaftlichen Zusammenhang. [6]

Nicht alle im ABAP/4 Dictionary definierten Tabellen befinden sich auch so in der Datenbank. So können im R/3 ABAP/4 Dictionary die folgenden Tabellentypen definiert werden:

- Transparente Tabellen
- Pool-Tabellen
- Clustertabellen

Bei transparenten Tabellen entspricht die Tabellendefinition im ABAP/4 Dictionary der Definition in der Datenbank. Nur auf die Daten transparenter Tabellen kann mit nicht-R/3-eigenen Mitteln zugegriffen werden. Alle Tabellen, die Kundendaten enthalten, sind transparente Tabellen. Diese Art von Datenbank versteht man im Sprachgebrauch unter einer normalen relationalen Datenbanktabelle (beispielsweise aus ACCESS oder ORACLE).

Mit relationalen Datenbanken können aus verwaltungstechnischen Gründen oftmals weniger Tabellen verwaltet werden, als für das R/3-System nötig wären.

Dieser Mißstand kann umgangen werden, indem man die Inhalte mehrerer R/3-Tabellen (Pool-Tabellen) in einer einzigen Datenbanktabelle zusammenfaßt.

Tabellen sollten dann in einem Tabellen-Pool aufgenommen werden, wenn

- der Datenbankzugriff nicht extern erfolgt
- der Zugriff hauptsächlich über den Primärschlüssel erfolgt
- es sich um eine kleinere Tabelle handelt und
- die Tabelle vollständig gepuffert ist, d.h. wenn sie vollständig in den Hauptspeicher geladen ist)

Ein Cluster ist eine Speichereinheit. Diese umfaßt einen zusammenhängenden Bereich, der aus einer Mehrzahl von Basisspeicherungseinheiten besteht.

Im R/3-System werden so

- mehrere Tabellen des AB/4 Dictionary zu Objekten zusammengefaßt und
- in einer einzigen Tabelle als Cluster an die Datenbank weitergeleitet

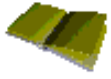
Bildet eine Tabelle zusammen mit anderen Tabellen oder mit mehreren Sätzen einer anderen Tabelle ein Objekt, werden sie zu Tabellen-Clustern zusammengefaßt.

Die Kenntnis dieser Unterschiede und das Wissen, welche R/3-Tabelle von welcher Art ist, ist wichtig für das Archivieren von Tabellen.

So ist das ADK in der Lage, transparente Tabellen mit einem hohen Faktor zu komprimieren, wohingegen Clustertabellen maximal mit einem Faktor 1,5 komprimiert werden können (lt. Aussage SAP-Berater).

Anhand der Buchungstabellen BKPF, BSEG, APQD und APQI sollen diese Komprimierungsmöglichkeiten visualisiert werden:

Die Tabelle mit den Belegköpfen (BKPF) ist eine transparente Tabelle.  
Die Tabelle mit den Belegpositionen (BSEG) ist eine Clustertabelle.  
Die Tabelle mit den Mappenkopfdaten (APQD) ist eine transparente Tabelle.  
Die Tabelle mit den Mappensteuerzeichen (APQI) ist eine transparente Tabelle.



### 9.3.4 Funktionsbausteine

ABAP/4-Programme sind in der Lage, Unterrouтины (Unterprogramme) zu verarbeiten. Zum Aufruf der Unterprogramme gibt es drei Möglichkeiten.

Unterrouтины können:

- im Programm selber gehalten sein
- aus einem anderen ABAP/4-Programm abgerufen werden oder
- als Funktionsbaustein definiert werden

Im folgenden soll das Prinzip der Funktionsbausteine kurz erläutert werden:

Sie unterscheiden sich von Unterrouتين durch:

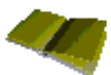
- eine deutlich fixierte Datenschnittstelle
- eine Testumgebung, in der die Funktionsweise auf Korrektheit und die Werte auf ihre Konvertierungsmöglichkeiten hin überprüft werden
- eine Funktionsbibliothek zur Verwaltung der Bausteine und
- eine Konzeption zum Behandeln von Ausnahmefällen

Bei den Funktionsbausteinen handelt es sich allgemein um ausgelagerte Unterprogramme eines Reports, die eine allgemeine Funktionsweise zur Lösung eines Problems darstellen. Da bestimmte Probleme in vielen Reports immer wieder auftauchen und wieder neu programmiert werden müßten, wurde die Lösung der Funktionsbibliothek gewählt. Wie in vielen anderen Sprache (z.B. C++, VB, JAVA), wo Bibliotheken über DLLs oder INCLUDEs eingebunden werden, wird auch in der ABAP/4-Umgebung auf eine Bibliothek zugegriffen. Das Einbinden fällt hier jedoch weg, da über das DDIC alle Programme, Module oder Funktionen zentral verwaltet werden.

Funktionsbausteine stehen in Funktionsgruppen, die bei dem Programmstart komplett zum Source-Code geladen werden.

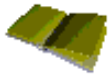
Die Funktionsbausteine werden über die Funktion CALL ... aufgerufen. Bei diesem Aufruf können alle notwendigen Parameter, wie z.B. welche Felder importiert oder exportiert werden, oder für welche Fehlerfälle eine Ausnahmebehandlung aktiviert werden soll, angegeben werden. Der Funktionsbaustein übernimmt die Steuerung zur Laufzeit und gibt sie nach erfolgreicher Verarbeitung zurück.

In der SAP R/3 Version 3.0D, die momentan bei der BAYER AG eingesetzt wird, existieren z.Zt. genau 39.324 Funktionsbausteine [Tabelle: V\_FDIR].



## 9.4 Enwicklung

Wenn man allgemein einen neuen Report anlegen möchte, muß man als Softwareentwickler unter SAP autorisiert sein. Diese Zertifizierung vergibt das SAP-CompetenceCenter der BAYER AG, welches als Verbindungsstelle zwischen der BAYER AG und der SAP AG in Walldorf fungiert, in Form eines 20stelligen numerischen Entwicklungsschlüssels. Er ist personenbezogen und identifiziert eindeutig. Nachdem einmalig dieser Schlüssel dem System bekannt gemacht worden ist, darf entwickelt werden. Jeder Entwickler erstellt in einer Entwicklungsklasse seine Programme. Vor dem Beginn der Entwicklung komplexer Servicelösungen sollte eine Entwicklungsklasse angelegt werden, da die Entwicklungsklasse als globales Objekt alle notwendigen Entwicklungsobjekte, wie z.B. Domänen, Datenelemente, Tabellen, Dynpros etc. zusammenfaßt. Eine Entwicklungsklasse umfaßt mehrere logisch zusammengehörende Entwicklungsobjekte. Diese können beispielsweise alle Objekte einer Anwendung sein. Eine Entwicklungsklasse selbst kann aber wiederum ein Entwicklungsobjekt sein. Eine komplette Entwicklungsklasse kann sehr komfortabel transportiert werden. Die Zuordnung einzelner Entwicklungsobjekte zu einer Entwicklungsklasse ermöglicht eine einheitliche Korrektur oder den Transport des gesamten Projektes. Diese ist für das KTW notwendig. Entwicklungsklassen sind in der Tabelle TDEVK verzeichnet. Auf der Ebene der Entwicklungsklasse wird entschieden, ob die Objekte überhaupt das KTW verwenden sollen. Ein Beispiel hierfür sind lokale private Objekte. Sie befinden sich in der Entwicklungsklasse \$TMP. Alle Entwicklungsklassen, die mit T oder \$ beginnen, unterliegen nicht dem Änderungsmechanismus des KTW. Das Integrationssystem und das Konsolidierungssystem werden pro Entwicklungsklasse angegeben.



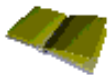
## 9.5 Testumgebung

Unter einer Testumgebung versteht man Testwerte, mit denen das Programm auf semantische Fehlerfreiheit getestet werden soll.

So müssen beispielsweise, um den Abstimmreport zu testen, geeignete Inputdateien, im vorliegenden Fall also BI-Schnittstellendateien, generiert werden.

Das kann auf zwei verschiedene Arten geschehen:

- Direktes Buchen
- Abspielen von Batch-Input-Mappen

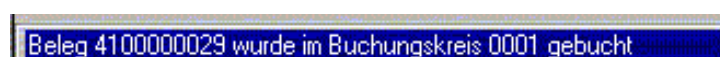


### 9.5.1 Direktes Buchen

Beim direkten Buchen wird Online eine Buchung im SAP-System durchgeführt.

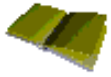
Dies geschieht über die schon bekannte Transaktion <FB01>. Die notwendigen Felder werden manuell gefüllt, und der Beleg wird gebucht.

Am Ende dieser Online-Buchung wird die vom System vergebene Belegnummer dem Buchenden in der Fußzeile des Bildschirms bekanntgegeben (s. Abb. 21):



## Abbildung 21 - Statuszeile FB01 nach erfolgreicher Buchung

Um sich aus dieser Buchung eine passende Testinputdatei zu generieren, muß auf einen Report zugegriffen werden, der diese Buchung anhand der Belegnummer identifiziert und in eine UNIX-Struktur umsetzt. Die daraus resultierende Ausgabedatei kann als Eingabedatei jedoch nur für die Semantikprüfung des Abstimmreports genutzt werden. Es gleicht einem SAP-SAP-Lauf und kann daher nur die generelle Funktionsfähigkeit des Reports nachweisen, nicht jedoch den Umgang mit bitweise verschobenen Feldinhalten o.ä. Problemen bei Fremdsystembuchungen.



### 9.5.2 Abspielen von Batch-Input-Mappen

Eine andere Möglichkeit ist, aus dem Remote-System (hier: MBS auf dem HOST) gebuchte Datensätze zu einer Datei zusammenzufassen, ins Entwicklungssystem zu transportieren und daraus BI-Mappen generieren zu lassen. Diese Mappen werden dann abgespielt, und die Belege in SAP gebucht. Die Datei, die die Daten für die Mappen lieferte, dient nun als Eingabedatei für den Report.

#### Exkurs ALE:

Das Application Link Enabling machte verteilte Geschäftsprozesse im System R/3 möglich. Die Kommunikation innerhalb von ALE läuft über Nachrichten. So ermöglicht ALE eine lose gekoppelte Synchronisation von physisch getrennten R/3-Systemen (z.B. Arbeitsfeldern), die jeweils eine eigene Datenbank besitzen (Grundprinzip s. Abb. 22). ALE führte schwach detaillierte Komponenten und entsprechende Schnittstellen für Anwendungsprogrammierung (APIs – Application Programmers Interface) in das System R/3 ein. Diese Schnittstellen ermöglichen jedoch nicht nur die Kopplung von R/3-Systemen, sondern auch eine Verbindung zwischen R/2- und R/3-Systemen sowie eine Verbindung zwischen R/3- und Fremdsystemen. ALE unterstützt sogar die Verbindung von R/3-Systemen mit unterschiedlichen Release-Ständen.

Die wichtigsten Komponenten des ALE sind:

- BO: Business-Objekt, besteht aus einem Kern, der die betriebswirtschaftliche Logik enthält. Dazu kommen Geschäftsregeln, Methoden, Attribute und Zugriffsebenen
- BOR: Business Objects Repository als Bestandteil des R/3-Repository, das die BO verwaltet
- BAPI: Business Application Programming Interface – über diese Schnittstelle kann von außen auf Geschäftsprozesse und Business-Objekte zugegriffen und so eine direkte Kommunikation zwischen Anwendungen unterschiedlicher Software-Anbieter ermöglicht werden.

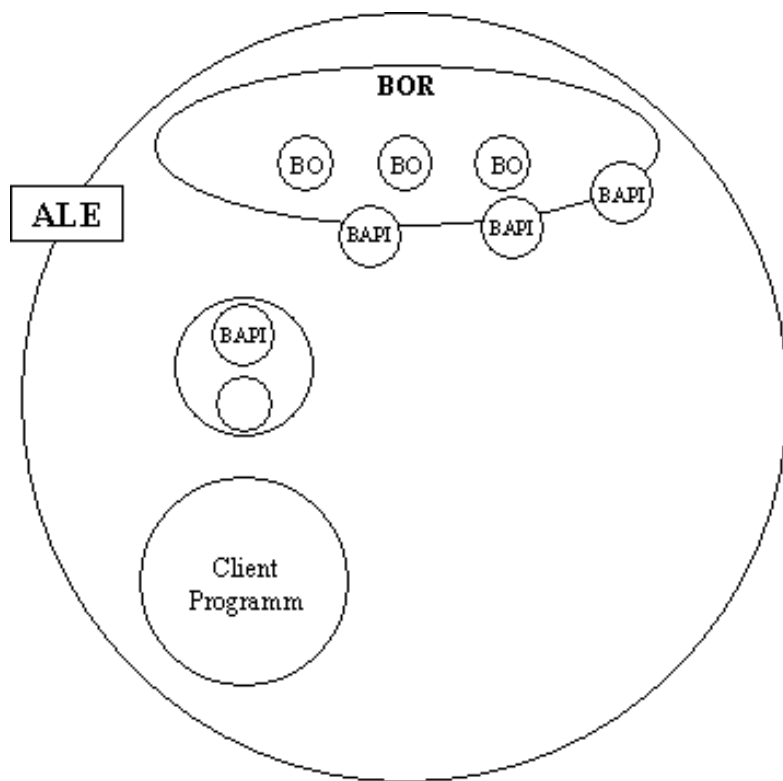
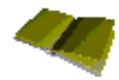


Abbildung 22 - ALE–Grobstruktur [9]

Ende Exkurs



# Der Abstimmreport

## Kurzbeschreibung

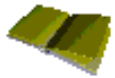
## Detaillierung des Reports

---

### 10 Der Abstimmreport

Der Report mit dem Namen ZFSAPRO9 wurde in der in Kapitel 9 beschriebenen Umgebung ABAP/4 auf einem Entwicklungssystem entwickelt und auf syntaktische und logische Korrektheit geprüft. Auf einem QS-System wurde das korrekte Übertragen des Transportes mittels KTW geprüft und auf dem Produktivsystem für das Rechnungswesen eingesetzt.

Im Folgenden wird die Struktur und die zugrundeliegende Idee des Reports erläutert.



#### 10.1 Kurzbeschreibung

Als Eingabedaten benutzt der Report eine Datei mit MBS-Buchungsdaten. Diese Datei wird nur einmal gelesen und dabei sequentiell abgearbeitet.

Als Ausgabedatei schreibt der Report eine weitere Datei, in die diejenigen Sätze zurückgeschrieben werden, deren Belegköpfe nicht im SAP-System gefunden wurden. Dies kann den Grund haben, daß die Buchung gelöscht wurde, aber auch, daß der Beleg noch in einer BI-Mappe hängt, die noch nicht oder fehlerhaft verarbeitet wurde.

Alte Belege, die ein Fälligkeitsdatum überschritten haben (Buchungsdatum), werden voraussichtlich im SAP-System nicht mehr auftauchen. Deshalb gibt es in den Selektionsparametern\* ein Feld, in das eingetragen werden kann, ab wieviel Tagen Überfälligkeit der Beleg nicht mehr berücksichtigt werden soll. Der Default dieses Wertes liegt bei 61 Tagen.

In einem weiteren logischen Block auf der Eingabemaske kann der Anwender eingeben, ob die nicht im SAP-System gefundenen Buchungen vor der Ausgabe sortiert werden sollen. Sollte dieser Fall gewünscht sein, so gibt es die Möglichkeit, nach verschiedenen Sortierkriterien zu unterscheiden.

Der Report besitzt noch eine weitere Funktionalität: Er kann als Diagnoseprogramm eingesetzt werden. In dieser Funktion prüft er, welche Buchungen aus der Schnittstellendatei gar nicht ins SAP-System gelangt sind. Die Feldprüfung wird zu diesem Zweck deaktiviert.

In diesem Fall gibt der Report nur die nicht gefundenen Buchungen aus, nicht jedoch Feldabweichungen und korrekte Belege. Dies war nicht die eigentliche Anforderung, sondern wurde als Funktionalität (quasi als Extra-Feature) später implementiert.

Im nachfolgend aufgeführten Struktogramm wird der Fall der eingeschalteten Feldabgleichsprüfung, also der Default-Fall mit vollem Funktionsumfang, zugrundegelegt.

Das Programm gliedert sich in mehrere Segmente:

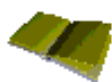


- Kopf mit Reportname und Informationen
- Datendefinitionen und Dateiaufbaudefinitionen
- Füllen einiger wichtiger interner Tabellen
- Lesen und Abarbeiten Satz für Satz aus der Eingabedatei

unter Umständen sequentielles Schreiben der Ausgabedatei  
eventuelles Feldabweichungsprotokoll ausgeben

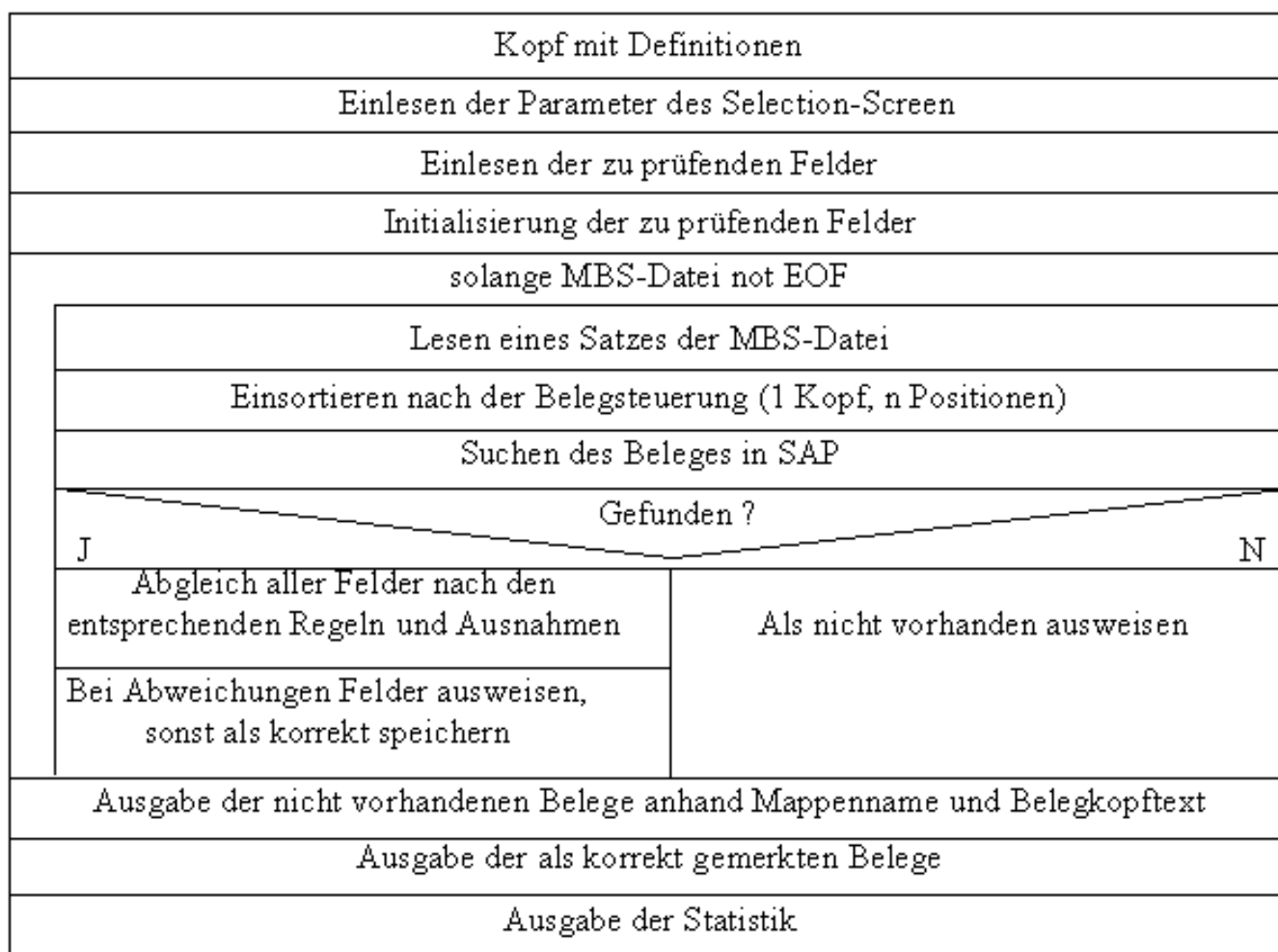
Füllen interner Tabellen für die Statistik und die Protokolle

- Ausgabe der restlichen Protokolle in die Liste
- Ausgabe der Statistik in die Liste



## 10.2 Detaillierung des Reports

Der Report ist in der SAP R/3 – Entwicklungsumgebung in der Sprache ABAP/4 geschrieben und nach folgendem Schema aufgebaut (s. Struktogramm 1):



### Struktogramm 1 - Struktureller Ablauf des Reports

*(Dieser Ablauf stellt das Schema mit vollständigem Funktionsumfang bei eingeschalteter Feldprüfung dar)*

Der Kopf enthält Daten über den Report wie den Autor, den Entwicklungsrechner, die Entwicklungsklasse, die

Art des Dateizugriffs (Read, Write, Read&Write) und den Reportnamen.

Der Definitionsteil definiert die internen Tabellen, die sich z.T. aus den Tabellenstrukturen der einzelnen Tabellen zusammensetzen. So gibt es beispielsweise die interne Tabelle UNIX\_TAB, die sich aus Feldern der BGR00, der BBKPF und der BBSEG zusammensetzt. Als Basis dient hierzu die BBSEG, und die ebenfalls benötigten Felder der BGR00 und der BBKPF werden hinzugefügt. Die Struktur der BBSEG wird mit Hilfe des Befehls INCLUDE STRUCTURE BBSEG hinzugefügt.

Grund für diesen Aufbau ist, daß in dieser Tabelle die einzelnen SA der MBS-Sätze aus der BI-Schnittstellendatei zu einem einzigen Datensatz zusammengefaßt werden. Ein solcher Daten-satz beinhaltet dann alle Informationen, die eine Belegposition identifiziert, und nach denen die Position im SAP-System gesucht werden kann.

Das Pendant dazu ist die Tabelle SAP\_TAB, die sich aus der Basistabelle BSEG und den benötigten Teilen der Tabelle BKPF zusammensetzt. Wird ein MBS-Beleg im SAP-System ausgemacht, wird der komplette Beleg aus der BSEG und der BKPF in dieser internen Tabelle zum Vergleich aufbereitet.

Beim Einlesen der Parameter des Selection-Screens wird dem Benutzer eine Bildschirmmaske angeboten, in der er die Namen der Eingabedatei und Ausgabedatei (jeweils mit Pfad) sowie das maximale „Alter“ des Beleges, ausgehend von seinem Buchungsdatum, eingeben muß (s. Abb. 23).

Es existieren Default-Werte zu diesen 3 Feldern, die jedoch geändert werden können. Jedoch ist der Eintrag in diesen Feldern obligatorisch.

Desweiteren kann angegeben werden, ob, und wenn ja nach was, die Liste der nicht vorhandenen Buchungen zu sortieren ist.

Bitte Parameter eingeben

Bitte physische Dateinamen angeben !

Name der Eingabedatei:

Name der Ausgabedatei:

Nicht vorhandene Belege werden auf Wiedervorlage gelegt,  
sofern sie nicht älter als die eingegebenen Tage sind.  
Diese werden dann in die Ausgabedatei geschrieben.

Bitte erlaubte Lebensdauer des Beleges in Tagen eingeben !

Wieviel Tage alt dürfen die Belege sein ?

Wonach sollen nicht vorhandene Belege sortiert werden?

NICHTS ☒

BKTX ☐

T\_DIFF ☐

Sollen die Felder auf Abweichung geprüft werden ?

Felder prüfen ☒

## Abbildung 23 - Eingabemaske des Reports

Bei der Initialisierung der zu prüfenden Felder werden alle Felder eines Datensatzes, die sowohl im MBS als auch im SAP enthalten sind, in interne Tabellen geschrieben. Dabei können dies Felder sein, die den gleichen (z.B. TCODE), aber auch die unterschiedliche Namen in den Strukturen haben (NEWKO und HKONT). Diese Felder werden in die interne Tabelle GEMEINSAME\_FELDER geschrieben.

Eine sehr wichtige Tabelle in diesem Zusammenhang ist die Systemtabelle DD03L. In ihr sind sämtliche Informationen aller Attribute aus allen im DDIC eingetragenen Relationen registriert. Die interne Tabelle GEMEINSAME\_FELDER nutzt diese Informationsquelle, indem sie sich aus Werten der Tabelle DD03L füllt. Aus der DDIC-Tabelle werden diejenigen Felder in die interne Tabelle geschrieben, die den gleichen Namen sowohl in einer der Tabellen aus (BGR00, BBKPF, BBSEG) als auch in einer der Strukturen aus (BSEG, BKPF) haben.

Abbildung 24 zeigt exemplarisch die Existenz des Feldes TCODE in den Beschreibungen der Strukturen BBKPF und BKPF.

Felder mit unterschiedlichen Namen wurden im Quellcode manuell an die Tabelle angehängen, da aus der DD03L die Beziehungen der Felder nicht hervorgehen.

**Data Browser: Tabelle DD03L** 7 von 7 Treffern

Tabelle Bearbeiten Springen Einstellungen Hilfsmittel Umfeld System Hilfe

Angezeigte Felder: 21 von 21 Feststehende Führungsspalten: 4 Listbreite 250

	TABNAME	FIELDNAME	AS4LOCAL	AS4VERS	POSITION	KEYFLAG	MANDATORY	ROLLNAME
<input type="checkbox"/>	BBKPF	TCODE	A	0000	0002			TCODE
<input type="checkbox"/>	BKPF	TCODE	A	0000	0015			TCODE_BKPF
<input type="checkbox"/>	EBKPF	TCODE	A	0000	0016			TCODE_BKPF
<input type="checkbox"/>	FUBKPF	TCODE	A	0000	0018			TCODE_BKPF
<input type="checkbox"/>	VBKPF	TCODE	A	0000	0017			TCODE_BKPF
<input type="checkbox"/>	UEBKPF	TCODE	A	0000	0017			TCODE_BKPF
<input type="checkbox"/>	ZFKBPF	TCODE	A	0000	0011			TCODE_BKPF

Abbildung 24 - Ergebnisbild einer DD03L-Anfrage

Es ist unvermeidlich, daß im Laufe der Übertragung („Ausblender“, Verbuchungsreport RFBIBL00) einige Felder ihren Inhalt verlieren, ohne daß manuell eingegriffen wurde.

Diese Felder stellen Ausnahmen im Abgleich dar, da sie ansonsten immer ausgewiesen würden. Um diese Felder nicht zu vergleichen, werden sie aus der internen Tabelle gelöscht.

Desweiteren gibt es noch Work-Area-Bereiche für die einzelnen SA, in die die einzelnen MBS-Daten übertragen werden. Work-Area-Bereiche sind interne Tabellen mit dem identischen Tabellenaufbau der DDIC-Tabelle. Damit wird sichergestellt, daß die eingelesenen Daten permanent zur Verfügung stehen.

Es existiert eine interne Tabelle für Belege, die nicht gefunden wurden, und dort für die Ausgabe am Ende festgehalten werden.

Nicht zu vergessen ist natürlich der Work-Area-Bereich der eigentlichen sequentiellen Datei mit den MBS-Daten. In diesen wird der Satz erst einmal eingelesen, bevor geprüft werden kann, zu welcher SA er gehört. Von dort aus wird er dann in die jeweilige Work-Area und in die interne Tabelle UNIX\_TAB geschoben, um

dort aufbereitet zu werden.

Diese Auszüge der Datendefinition sollen exemplarisch für den wichtigsten Bereich fast jeder entwickelten Software stehen. Der Aufwand der Erstellung und vor allem der späteren Pflege von Programmen hängt in erster Linie davon ab, wie gut und wie Strukturiert die Datendefinition ist.

Es gibt noch eine Vielzahl von Zählern und Flags für die unterschiedlichsten Dinge (Hilfsvariablen), die in diesem Bereich initialisiert werden, auf die jedoch hier nicht weiter eingegangen werden soll.

Eine weitere interne Tabelle (BKPF\_POOL) speichert die abhängig vom Buchungsdatum in Betracht kommenden Belegköpfe. In ihr werden alle im System verfügbaren Belegkopfsätze der aktuellen und der letzten Periode gesammelt.

Dies hat folgenden Grund:

Nach jedem MBS-Beleg, der über die Eingabedatei eingelesen wird, erfolgt ein Zugriff auf die DDIC-Tabelle BKPF, um anhand des Belegkopfes und des Geschäftsjahres aus über 2 Millionen Belegen die entsprechende FI-Belegnummer zu ermitteln. Da diese Tabelle nicht vollständig gepuffert ist und das bei einigen tausend MBS-Belegen dementsprechend langwierig wäre, werden die in Frage kommenden Belegköpfe intern abgelegt. Dieses Verfahren erhöht die Zugriffsgeschwindigkeit auch noch deswegen, weil die interne Tabelle nach Belegkopftext sortiert wird, und daher ein binäres Suchen möglich ist. Der Faktor, um den sich diese Zugriffe verschnellern, liegt im vierstelligen Bereich. Einziger Nachteil dieses Verfahrens ist, daß für diese interne Tabelle zusätzlich Hauptspeicher nötig ist. Dank des Speicherverwaltungskonzept von SAP und der guten Hardware ist diese interne Tabelle zu verschmerzen.

Besonders erwähnenswert ist jedoch die Verwendung der „FIELD-SYMBOLS“. FIELD-SYMBOLS stehen immer in eckigen Klammern. Auf dieser Variablendefinition beruht das komplette Feldabgleichverfahren. Mit der Anweisung FIELD-SYMBOLS wird ein Feldsymbol vereinbart. Zur Laufzeit wird dem Feldsymbol durch die Assign-Anweisung ein konkretes Feld zugewiesen. Alle Operationen, die mit dem Feldsymbol durchgeführt werden, werden dann mit dem ihm zugewiesenen Feld durchgeführt. Feldsymbole können verweisen auf: Felder, Feldleisten und Strukturen, die mit der TABLES-Anweisung deklariert worden sind. Eine Anwendung ist, dem Feldsymbol den jeweiligen Feldnamen zuzuweisen, der verglichen werden soll

```
(ASSIGN `TCODE` to <L_BBSEG>).
```

Bei der Eröffnung der Schleife wird die Eingabedatei, deren Name über den Selection-Screen eingegeben wurde, zum Lesen geöffnet. Die im Struktogramm angegebene Bedingung EOF ist so realisiert, daß ein Satz gelesen wird. Handelt es sich hierbei um den letzten Satz der Datei, so wird beim nächsten Lesen der Systemreturncode ungleich Null. Das ist das SAP-interne Zeichen, daß eine sequentielle Datei keinen weiteren Satz mehr besitzt.

Jeder Schleifendurchlauf beginnt mit einem sequentiellen Lesen eines weiteren Satzes. Wenn kein ungelesener Satz mehr in der Datei steht, ist der Systemreturncode Null, und die Verarbeitung wird mit der Anweisung EXIT verlassen.

Das Einsortieren nach der Belegsteuerung (1 Kopf, n Positionen) erfolgt anhand der SA des MBS-Satzes. Der Anfang eines jeden Satzes beginnt mit einer einstelligen SA, die 0, 1 oder 2 sein kann.

Die SA 0 leitet einen Mappensatz ein, also einen Satz der den RFBIBL00 veranlaßt, eine neue Mappe zu eröffnen.

Die SA 1 leitet eine neue Transaktion ein, und damit einen neuen Beleg. Alle FI-Belege haben einen FB-Tcode, z.B. FB01. In diesem Satz ist ebenfalls der wichtige Belegkopftext vorhanden.

Die SA 2 leitet einen Positionssatz ein. Da das sowohl ein Belegsegment als auch ein Steuerbetragssatz sein kann, ist das zweite Feld eines jeden Satzes mit der SA 2 der Tabellename. Hier steht explizit drin, ob es sich um einen Satz aus der BBSEG oder der BBTAX handelt.

Jede dieser vier Satzarten (BGR00, BBKPF, BBSEG, BBTAX) hat seinen eigenen Work-Bereich. So wird nach erfolgreicher Identifikation der SA der Satz in seinen jeweiligen Work-Bereich kopiert.

Anschließend erfolgt ein Kopieren aller Felder der Work-Bereiche in die interne Tabelle UNIX\_TAB.

Um eine aussagekräftige Statistik zu erhalten, wird bei Auftreten einer jeden SA der entsprechende SA-Zähler

hochgezählt.

Im folgenden wird sich mit dem Suchen eines Beleges in SAP beschäftigt.

Der Belegkopf des eingelesenen MBS-Satzes wird in der internen Tabelle BKPF\_POOL gesucht. Das erste Problem tritt auf, wenn es sich um einen Beleg aus der vor-vor-Periode handelt. Dann ist er, obwohl eventuell im SAP-System vorhanden, nicht in der internen Tabelle zu finden, und wird deswegen als nicht gefunden ausgewiesen.

Dieser Fall tritt jedoch selten ein, da die Belege so gut wie nie solange in einer Mappe hängen, daß die Periode abläuft.

Das nächste Problem tritt jedoch auf, wenn mehrere Belege den gleichen Belegkopftext angeliefert haben.

Dann findet der Report immer den ersten Beleg bei jedem Suchen mit dem Belegkopftext.

An einem gestellten Beispiel soll dieses Problem verdeutlicht werden:

MBS (Daten, die aus dem Vorsystem kommen)

*Betrag Belegkopf*

100,00 ABCDEF

200,00 ABCDEF

BKPF\_POOL

*Belegkopf Belegnummer*

ABCDEF 4711

ABCDEF 4712

BSEG

*Belegnummer Betrag*

4711 100,00

4712 200,00

Offensichtlich hat SAP die Belege sauber verbucht, und jedem eine Belegnummer gegeben. Ein Suchen mit den MBS-Daten würde nun folgendes Ergebnis bringen:

Ein Satz mit einem Belegkopftext ABCDEF und dem Betrag 100,00 soll im SAP-System gefunden werden.

Mit Hilfe des Belegkopfes wird in der BKPF\_POOL gesucht und gefunden. Die gefundene Belegnummer ist 4711. Mit dieser wird in der BSEG gesucht und der Betrag 100,00 gefunden. Bis hierhin ist also alles in Ordnung.

Nun wird der zweite MBS-Satz gelesen. Der Belegkopftext ist ABCDEF und der Betrag 200,00. Wieder wird in der BKPF\_POOL der Belegkopftext gesucht und gefunden. Da immer der erste Eintrag genommen wird, ist die gefundene Belegnummer wieder 4711. Daher wird eine Abweichung ausgewiesen: MBS stellt den Wert 200,00, SAP den Wert 100,00.

Die geschilderte Problematik ist typisch für Daten, die aus dem MBS-System kommen. Bei der Umsetzung dieser Buchungen kann es vorkommen, daß der Belegkopftext nicht eindeutig ist. Einfacher ist das Abgleichen von anderen Arbeitsfeldern, da dabei immer ein eindeutiger Belegkopf vorhanden ist.

Allgemein gilt:

Es wird der Belegkopftext in der internen Tabelle BKPF\_POOL gesucht, der im MBS-Daten-Satz mit der SA 1 enthalten ist. Wird dieser gefunden, wird weiterverarbeitet. Wenn nicht, wird geprüft, ob das Buchungsdatum des Beleges älter ist als die erlaubten Tage, die im Selection-Screen angegeben werden können.

Dies hat den Sinn, daß Belege, die älter als beispielsweise zwei Monate (default) sind, aller Voraussicht nach nicht mehr in BI-Mappen hängen, sondern verloren oder „zerrupft“ sind. „Zerrupft“ sind Belege, die nicht mehr in Reihenfolge stehen, sondern wo Positionen manuell ins SAP-System eingebucht werden. Diese Daten müssen nicht unbedingt weiter mitgeschleppt werden.

Ist der Beleg nicht älter, so wird er in die interne Tabelle NICHT\_DA geschrieben, und der ganze Satz wird in die Ausgabedatei geschrieben. Diese wird vor dem nächsten Lauf mit den neuen MBS-Daten zu einer neuen Eingabedatei auf Betriebssystemebene gemerged.

Das Suchen einer Belegposition nach erfolgreichem Suchen des Belegkopfes beginnt mit der Abfrage, ob es sich um ein Belegsegment oder ein Steuersatz handelt. Dies läßt sich leicht feststellen, da die SA 2 als zweites Feld den Tabellenstrukturnamen beinhaltet.

Wenn es sich nicht um einen Steuersatz handelt, wird nach ihm in der BSEG mit den Kriterien Buchungskreis, Buchungszeile, Belegnummer, Geschäftsjahr, Buchungsschlüssel und der BuchungszeilenID gesucht.

Letztere darf im gegebenen Fall nicht den Wert 'T' haben, da der für TAX steht. Die Buchungszeile ist ebenso notwendig, da es mehrere Positionen (je nach Fall mehrere 100) für ein Beleg geben kann. Somit würde das Problem auftreten, daß nicht eindeutig identifiziert werden kann, welche Position gemeint ist.

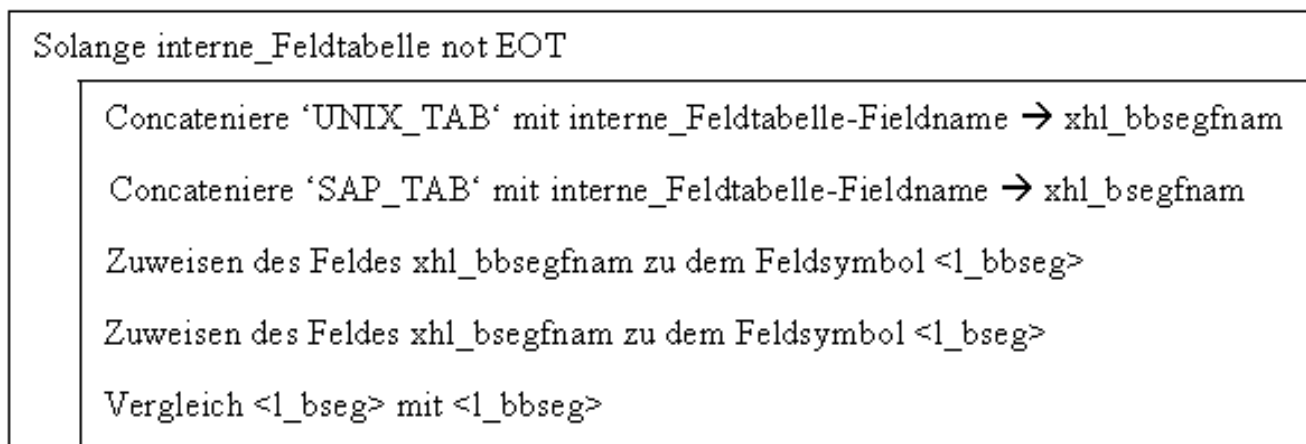
Wenn es sich um einen Steuersatz handelt, wird der passende Belegsegmenteintrag anhand der Felder Belegnummer, Mehrwertsteuerkennzeichen, Geschäftsjahr, Buchungskreis, BuchungszeilenID und Buchungsschlüssel gesucht. Hier ist die Zeilennummer nicht notwendig, da sich jeder Steuersatz eindeutig anhand von Mehrwertsteuerkennzeichen und Buchungsschlüssel identifizieren läßt. Dieses Tupel kommt nur einmal pro Beleg vor [SAP-Technik].

Sollte die Belegposition nicht gefunden werden, so wird eine Fehlermeldung mit dem Text 'Kein BSEG zu BKPF-BKTX gefunden !' und 'BKTX: ' mit dem Belegkopftext des nicht gefundenen Belegs anstelle einer Abweichung in die Report-Liste geschrieben.

Im Fall, daß zu dem Belegkopftext der passende Belegkopf und ebenso die gesuchte Belegposition gefunden wird, beginnt der Vergleich über alle vorhandenen Felder.

Wieder wird zwischen BBTAX und BBSEG unterschieden, da die Struktur BBTAX weit weniger Felder hat als die Struktur BBSEG.

Struktogramm 2 zeigt die allgemeine Logik des Feldvergleiches:



## Struktogramm 2 - Ablauf der Feldvergleiche MBS-SAP

Handelt es sich um einen Satz der BBTAX, so werden die Felder der internen Tabelle TAX\_FELDER für den aktuellen Satz nach dem oben gezeigten Schema verglichen. Davor müssen diese manchmal noch aufbereitet werden, und zwar wenn ein Feld, das in der BSEG als Typ P definiert ist, mit einem Feld aus den MBS-Daten verglichen werden soll, dessen Inhalt nicht in ein Feld vom Typ P konvertiert werden kann. [P = gepackt]

Nachdem die Schleife beendet ist, werden noch Felder mit ungleichem Namen verglichen.

Handelt es sich hingegen um einen Satz der BBSEG, so beginnt das Feldvergleichen mit den Feldern der internen Tabellen GEMEINSAME\_FELDER und UNGLEICHE\_FELDER ebenfalls jeweils nach dem oben gezeigten Schema. Nachdem die Schleife beendet ist, werden Felder mit ungleichem Namen verglichen.

Am Schluß der Feldabgleiche werden noch Felder verglichen, deren Namen sich erst dynamisch zur Laufzeit

ergeben, abhängig von dem aufgetretenen Buchungsschlüssel.

Bezüglich der Aufbereitung der Felder werden die gleichen Unterprogramme benutzt.

Bei Abweichungen eines Feldes erfolgt das Ausweisen der Abweichung mit dem tatsächlichen Wert, dem verbuchten Wert, der Belegnummer, der Buchungszeile, des Belegkopftextes und des Mappennamens (s. Abb. 25).

Map.-name Belegnr	Buwei	Feld	Belegkopftext MBS-Daten	SAP-Daten
ZFS_101SZ 4000000059	001	AUFNR WRBTR	SZ/60030600/27818 CX043069777 5123326.32	BATCH  CX0369904160 326,32
ZFS_101SZ 4000000060	001	AUFNR PERNR	SZ/60030600/27819 CX044977626 04977626	BATCH  CX0369904160 03711363
ZFS_101SZ 4000000061	001	AUFNR GSBER	SZ/60030600/27820 CX043636914 /	BATCH  CX0369904160 0200
ZFS_101SZ 4000000062 4000000062	001 T003	AUFNR TXJCD	SZ/60030600/27821 CX042489543 toller Witz	BATCH  CX0369904160

**Abbildung 25 - Ausgewiesene Belegabweichungen**

Durchläuft ein Beleg alle Feldprüfungen ohne eine einzige Abweichung, so wird er zu den korrekten Belegen in die entsprechende interne Tabelle geschrieben.

Jedoch werden nicht alle Abweichungen ausgewiesen oder als Fehler registriert. Einige Felder, die als Fehler in jedem Beleg wieder auftauchen würden, werden von Anfang an aus der Betrachtung herausgenommen. Die Rede ist hier von denjenigen Fehlern, die beim manuellen Abspielen der BI-Mappen nicht verändert werden können, oder die zwangsläufig falsch sein müssen. Unter solche Felder fällt beispielsweise der Username. Der User, der die Daten aus MBS gebucht hat, ist in den seltensten Fällen identisch mit dem, der die entsprechende BI-Mappe abspielt. Diese Abweichung auszuweisen, und damit einen Beleg für nicht identisch zu erklären, würde doch etwas am Ziel vorbeigehen.

Das gleiche gilt für die Belegnummer. Im MBS-System wurde eine Belegnummer vergeben, als der Beleg verbucht wurde. Da dieser Beleg im SAP-System völlig neu verbucht wird, wird das Feld BELNR vom „Ausblender“ auf NO-DATA (/) gesetzt (s. Abb. 26).

Feldname	Beschreibung	Feldinhalt
STYPE	Satztyp	1
TCODE	Transaktionscode	FB01
BLDAT	Datum	29031999
BLART	Belegart	KA
BUKRS	Buchungskreis	0001
BUDAT	Datum	29031999
MONAT	Geschäftsmonat	03
WAERS	Währung	DEM
KURSF	Umrechnungskurs	/
BELNR	Belegnummer	/
WWERT	Datum	29031999
XBLNR	Referenz	
BUORG	Übergreifende Nr	
BKTX	Belegkopftext	D0154000011630000173IMYAL
PARGB	PartnerGsber.	/
AUGLV	Ausgleichsvorgang	/
VBUND	Partnergesellschaft	/
XMWST	Steuer rechnen	/
DOCID	Dokumentart	/
BARCD	Archiv-Dokument-Id	/
STODT	Datum	/
SENDE	Satzendekennzeichen	/
ZZANRMFI19	Mikrofilmnummer	
ZZFKRWFKZ	Workflow-Kennzeichen	

Abbildung 26 - Inhalt eines BBKPF nach dem Ausblenden

Da in der SAP-Tabelle natürlich nach dem Buchen eine gültige Belegnummer existiert, unter der die Segmente zu finden sind, müßte auch hier das Feld als Abweichung ausgewiesen werden. Da diese Abweichung jedoch unvermeidbar ist, werden diese und analoge Fälle von der Betrachtung ausgeschlossen.

Im Hinblick auf die hausinterne Revision wird jedoch zu Beginn einer jeden Report-Liste auf die angesprochenen Fälle hingewiesen.

Abbildung 27 zeigt diesen „Prolog“, der jedoch nur bei aktiviertem Feldabgleich erscheint.



Dieser Report weist Abweichungen zwischen BBSEG und BSEG Buchungen auf. Dabei werden jedoch einige Fälle absichtlich überlesen, da sie wegen SAP-eigener Logik nicht handlebar sind. Dieses sind folgende Fälle:

- 1) wenn BBSEG-ZUONR auf NO-DATA steht, ist der Satz nie fehlerhaft
- 2) BSEG-KOSTL wird nur ab 3. Stelle geprüft gegen BBSEG-KOSTL 1. - 8. Stelle
- 3) Das Feld MONAT wird nicht geprüft
- 4) Das Feld HKONT fallweise gegen 1 von 2 Feldern geprüft
- 5) Wenn das BBSEG-Feld auf NO-DATA steht, und das BSEG-Feld 0 oder initial ist, ist die Prüfung fehlerlos.
- 6) Die Prüfung auf das Feld ZUONR ist erfolgreich, wenn das BBSEG-Feld NO-DATA oder initial ist.
- 7) Die Prüfung auf das Feld VBUND ist erfolgreich, wenn das BBSEG-Feld NO-DATA oder initial ist.
- 8) Wenn das Feld BBSEG-DMBTR NO-DATA und BSEG-DBBTR = BSEG-WRBTR ist, ist diese Prüfung o.k.
- 9) Wenn das Feld BBTAX-FWSTE NO-DATA und BSEG-DBBTR = BSEG-WRBTR ist, ist diese Prüfung o.k.
- 10) Die Felder MONAT, HKONT, WWERT werden nicht 1:1 verglichen.
- 11) Das Feld USNAM wird nicht verglichen
- 12) Das Feld KURSF wird nicht verglichen

**Warnung: Es werden nur Belege von heute bis letzte Periode genommen.**

#### Abbildung 27 - Ausnahmenbekanntmachung zu Beginn einer Liste

Nachdem die Eingabedatei sequentiell gelesen und verarbeitet worden ist, erfolgt die Ausgabe. Während der Feldprüfungen wurden die internen Tabellen für korrekte und für nicht vorhandene Belege gefüllt. Diese werden nun ausgegeben.

Zuerst werden die nicht im System gefundenen Belege mit ihren Köpfen und Mappennamen in drei Kategorien aufgelistet.

In der ersten Kategorie werden die Belege aufgelistet, bei denen das Buchungsdatum noch keine drei Tage alt ist.

Die beiden anderen Kategorien richten sich nach den Benutzereingaben. Dort wurde eingegeben, ab welchem Alter der Beleg nicht mehr auf Wiedervorlage gespeichert werden sollte. Der Default liegt bei 61 Tagen (zwei Monaten). So würde die zweite Kategorie Belege zwischen drei und 61 Tagen, die dritte ? 61 Tagen enthalten.

Die Abbildungen 28-30 zeigen solche Ausgaben, die aus einer Testdatenreihe auf dem Entwicklungssystem resultierten.

**Nicht vorhandene Belege < 3 Tage**

Mappennummer	Belegkopftext	USER
Kein fehlender Beleg.		

**61 Tage >= nicht vorhandene Belege >= 3 Tage**

Mappennummer	Belegkopftext	USER
ZFS_101SZ	SZ/60030600/03060 BATCH	FI-BATCH-S 46 Tage

**Abgelaufene Belege > 61 Tage**

Mappennummer	Belegkopftext	USER
Kein fehlender Beleg.		

**Abbildung 28 - Listenanzeige der nicht vorhandenen Belege**

Danach werden die als korrekt eingestuften Belege ausgewiesen.

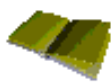
**Korrekte Belege**

Belegnummer	Mappenname	Belegkopftext
4100000071	ZFK001AF11_1	D0154000014100000071IMYAL

**Abbildung 29 - Listanzeige der korrekten Belege**

Abschließend wird noch eine kleine Statistik ausgegeben, wieviel Sätze welcher SA gelesen und verarbeitet wurden.

BGR00 gelesen:	40		
BbKPF gelesen:	481	davon nicht gefunden:	184
BbSEG gelesen:	715		
BBTAX gelesen:	29		

**Abbildung 30 - Statistik der gelesenen Belegsätze**

# Hauptprobleme

## Programmlogik

## Programmiertechnisch

## Testumgebung

## Archivierung + Reorganisation

Der Report wurde in der Entwicklungsumgebung des eingesetzten SAP R/3 – Systems für den Bereich FI unter der Verwaltung der Abteilung Informatik der BAYER AG erstellt.

Da die Arbeit auf einer konkreten Problematik basiert, mußte die Systemlandschaft, die Konventionen und die verschiedenen Bereiche des SAP R/3 – Systems kennengelernt und berücksichtigt werden.

Während der Entwicklung des Reports tauchten an den verschiedensten Stellen Probleme verschiedenster Natur auf. Einige waren ABAP/4 – Probleme, andere ergaben sich aus der Unkenntnis FI-interner oder auch übergreifender Zusammenhänge, die erst einmal evaluiert werden mußten.

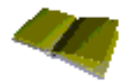
In einem Unternehmen wie der BAYER AG gibt es sehr komplexe Zusammenhänge, so daß bis zum Ende dieser Arbeit bei weitem noch nicht alle Fragen bis ins Detail geklärt sein können.

Die für das Erstellen der Arbeit relevanten Problemfelder sollen im folgenden kurz aufgeführt werden;

hierfür ist notwendig, zwischen zwei verschiedenen Problemfeldern zu unterscheiden:

Das eine sind Probleme, die sich in mangelhaften ABAP/4-Kenntnissen begründen lassen, das andere sind Logikprobleme, die sich aus der Aufgabenstellung und den zu prüfenden Daten ergeben.

Die am schwierigsten zu behabenden Probleme sind semantischer Natur, also das zweite Problemfeld.



## **11.1 Programmlogik**

Einen Report auf syntaktische Fehlerfreiheit zu optimieren, ist relativ trivial. Dank der guten Syntaxprüfung und der guten Onlinehilfe reichen oft rudimentäre Programmierkenntnisse aus, um einen ABAP/4-Report ausführbar zu machen.

Die echten Probleme tauchten im Verständnis der SAP R/3 – Logik und der vorhandenen Systemkonfiguration auf.

Je nach Berechtigungen kann jeder sich Buchungsbelege über das Rechnungswesenmodul anzeigen lassen, doch an den wenigsten Stellen der Literatur sind die beiden Buchungstabellen, in denen die Belegdaten abgespeichert werden, BSEG und BKPF, namentlich erwähnt.

Wenn die Tabellennamen der wichtigsten Relationen erst einmal bekannt sind, muß die Logik des Belegprinzips der SAP implementiert werden. Ein Beleg ist mit Hilfe seiner Belegnummer in beiden Tabellen wiederzufinden. Doch wie die beiden Tabellen miteinander verknüpft sind, und welche Logik hinter Belegsegmenten und Belegsteuersegmenten liegt, ist nicht offensichtlich. Diese Fragestellung findet sich ebenfalls im Bereich des Codings wieder.

Probleme wie diese gehören jedoch zu den angenehmeren, da sie relativ offensichtlich sind, und den Entwickler davon abhalten, mit dem Programmieren anzufangen.

Sobald diese Problemtypen erfolgreich überstanden sind, wird der erste Testlauf mit einigen Produktivdaten durchgeführt. Das Ergebnis ist unter Umständen frustrierend, da das System sich mit einem run-time-error

meldet, obwohl die verglichenen Felder den gleichen Namen und denselben Inhalt haben müßten. Ein solcher Fall kann auftreten, wenn ein Betragsfeld verglichen wird. Im SAP – System ist es vom Typ P und hat die Länge 13 mit 2 Nachkommastellen, also nach außen die Länge 16 ( 13 + 2 + 1 Trennzeichen). Das aus dem MBS-System angelieferte Feld ist hingegen vom Typ CHAR und hat 16 Stellen reserviert, in denen der Feldinhalt des SAP-Feldes u.U. bereits fertig aufbereitet steht. Doch eine Garantie dafür hat man nicht.

Abbildung 31 zeigt, wie dieses Feld in der Tabelle DD03L hinterlegt ist:

	TABNAME	FIELDNAME	INTTYPE	INTLEN	DATATYPE	LENG	DECIMALS
<input type="checkbox"/>	BBSEG	WRBTR	C	000016	CHAR	000016	000000
<input type="checkbox"/>	BSEG	WRBTR	P	000007	CURR	000013	000002

Abbildung 31 - Felddefinition eines Betragsfeldes BBSEG und BSEG

Der angelieferte Wert kann beispielsweise folgenden Aufbau haben (s. Tabelle 3):

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		2	2	6	.	5	4	2	.	1	2	5	,	0	0	
	0	0	0	0	0	0	0	0	0	6	5	,	0	0		
	0	0	0	0	0	0	0	6	8	6	0	0	3			

Tabelle 3 - Beispiele für MBS-Daten verschiedener Landeswährungen

Der Inhalt des Feldes ist scheinbar so unsicher, daß keine einheitliche Verarbeitung in der Form

```
IF Feld1 = Feld2.
  Anweisung1.
ELSE.
  Anweisung2.
ENDIF.
```

möglich ist. Die Struktur hängt vom Währungskennzeichen und den Nachkommastellen ab. Es liegt also eine Logik dahinter, doch die muß jedesmal erst über einen Algorithmus in eine mit dem BSEG-Feld vergleichbare Feldstruktur umgewandelt werden. Genau so kann es passieren, daß die Tausendertrennzeichen anstatt als Punkt als Komma dargestellt werden. Wenn ein solch angeliefertes Feld mit einem Betragsfeld verglichen wird, kommt es zum Laufzeitfehler, da die Kommata nicht als Trennzeichen interpretiert werden können. SAP erkennt, daß an dieser Stelle ein Betragsfeld mit einem CHAR-Feld verglichen wird, das nicht in ein Betragsfeld konvertierbar ist, und verweigert den Vergleich. Dieses Verhalten äußert sich in einem Laufzeitfehler.

Ein noch trickreicheres Phänomen trat mit folgender Meldung zum Vorschein (s. Abb. 32).

ZFS_1011K	1K/60030610/07623	BATCH	
4000000000	001 KOSTL	21625420	21625420

Abbildung 32 - Fehlerhaftes Ausweisen eines Feldes in der Report-Liste

Zwei auf den ersten Blick vollkommen identische Felder werden als voneinander abweichend ausgewiesen.

Ein Blick in die Schnittstellendatei und in die Buchungstabelle BSEG zeigt, daß der Inhalt beider Felder wirklich gleich zu sein scheint. Hier scheint eine genauere Betrachtung der Felder angebracht. Dies läßt sich über die Tabelle DD03L realisieren. In dieser sind alle Felder aus allen im SAP R/3-System verfügbaren Tabellen gespeichert.

Tabelle 4 zeigt das Ergebnis einer Prüfung:

	TABNAME	FIELDNAME	INTTYPE	INTLEN	DATATYPE	LENG	DECIMALS
<input type="checkbox"/>	BBSEG	KOSTL	C	000010	CHAR	000010	000000
<input type="checkbox"/>	BSEG	KOSTL	C	000010	CHAR	000010	000000

**Tabelle 4 - Feldaufbauvergleich zweier gleichnamiger Felder**

Nirgendwo ist ein Hinweis zu finden, was an den Feldern unterschiedlich sein könnte.

Eine letzte Möglichkeit, die Felder noch zu durchleuchten, ist der in die Entwicklungsumgebung integrierte Debugger

(s. Abb. 33).

Variablen

Gemeinsame_Felder-Fieldname	KOSTL		R
<L_Bbseg>	21625420		R
<L_Bseg>	0021625420		R
Sap_Tab-BeInr	4000000000		R

COMPUTE=FIXPOINT

**Abbildung 33 - Feldvergleich zur Laufzeit des Reports per Debugger**

Dieser weist die interne Behandlung der Feldinhalte aus. Im gegebenen Fall zeigt er an, nach welchen Regeln SAP scheinbar intern ein Feld aufbereitet.

Das betrachtete Feld hat den Namen KOSTL. Dahinter versteckt sich der Klartext Kostenstelle. Eine Kostenstelle setzt sich innerbetrieblich (bei BAYER) aus zwei Stellen Werksbezeichnung und fünf Stellen Kostenstelle mit ergänzender Null. Die betriebliche Länge einer Kostenstelle beträgt also acht Stellen. SAP liefert von Haus aus das Kostenstellenfeld zehnstellig aus.

Das Feld KOSTL ist als Feld der gleichnamigen Domäne angelegt. In der Beschreibung der Domäne sind Ausgabeeigenschaften wie Länge und Konvertierung definiert.

Abbildung 34 zeigt, was im System zu der Domäne angezeigt wird:

Ausgabeeigenschaften

Ausgabelänge	10
Konvert.-Routine	ALPHA

**Abbildung 34 – Eigenschaft der Domäne KOSTL**

Unter Konvertierungsroutine ist die Routine ALPHA angegeben. Das Feld verweist auf einen Funktionsbaustein, der den Feldinhalt prüft. Wenn dieser Feldinhalt aus Ziffern besteht, so konvertiert der Baustein das Feld in ein Feld von 100 Stellen mit dem bisherigen Feldinhalt rechtsbündig, aufgefüllt mit Nullen.

Im aktuellen Fall bekommt das DDIC-Feld KOSTL beim Buchen den Wert 21625420. Mit diesem Wert

durchläuft die Domäne den Funktionsbaustein:

CONVERSION\_EXIT\_alpha\_INPUT.

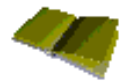
Dieser konvertiert den Wert in das zehnstellige Feld 0021625420, und das ist der Wert, der im Debugger ausgewiesen wird.

Da bei einem byteweisen Vergleich 0021625420 ungleich 21625420 ist, wird der Fehler ausgewiesen.

Bausteinbeschreibung:

Die ALPHA-Konvertierung wird besonders bei Kontonummern verwendet. Bei der Konvertierung vom externen in das interne Format wird geprüft, ob die Eingabe in Feld INPUT rein numerisch ist, also nur aus Ziffern, möglicherweise gefolgt von Leerzeichen, besteht. Wenn ja, dann wird die Ziffernfolge rechtsbündig in das Ausgabefeld OUTPUT gestellt, und dieses wird links mit Nullen ('0') aufgefüllt. Anderfalls wird die Zeichenfolge von links nach rechts in das Ausgabefeld übertragen, und dieses wird rechts mit Leerzeichen aufgefüllt.

Solche und ähnliche Fälle sind sehr schwer auszumachen, und erschweren die Arbeit und das saubere Vergleichen natürlich gewaltig.



## 11.2 Programmiertechnisch

Wie bereits angedeutet, ist es selbst mit rudimentären Kenntnissen möglich, einen ABAP/4-Report erfolgreich zu schreiben. Wenn jedoch einige Spezialfälle auftreten, die nicht zum Alltagsgeschäft des Entwicklers gehören, ist einiges an Recherche notwendig. Im Folgenden sollen einige wenige Probleme exemplarisch aufgegriffen werden.

SAP ist ein System, das auf Tabellen basiert, und der ABAP/4-Befehlssatz ist im wesentlichen auf Tabellenverarbeitung ausgerichtet.

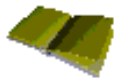
Im aktuellen Fall war jedoch ein Ein- und Auslesen von sequentiell angeordneten Sätzen aus einer Datei gefragt. Es mußte also nicht mehr auf SAP-Tabellen, sondern auf das Betriebssystem UNIX zugegriffen werden, und statt einer Tabelle mußte nun eine SAM\*-Datei ausgelesen werden. ABAP/4 stellt natürlich für diese bekannte Schnittstelle Befehle zur Verfügung, allerdings stehen die nicht so schnell auffindbar wie die Tabellenverarbeitungsbefehle, die dauernd benötigt werden.

Nach dem Einlesen und Einsortieren der MBS-Sätze in die entsprechenden internen Tabellen mußten die entsprechenden Buchungszeilen in den Buchungstabellen BKPF und BSEG gefunden werden.

Es kam die Fragestellung auf, wie es möglich sein könnte, einem beliebigen Belegkopf die passenden Segmente herauszusuchen. Bei normalem SQL wäre das Problem mit Hilfe eines trivialen INNER..JOIN zu lösen. Der ist jedoch in ABAP/4 nicht implementiert !!!

Mit einem READ...BINARY wurde der gesuchte Belegkopfsatz in der BKPF schnell gefunden. Mit Hilfe der gefundenen Belegnummer mußten mögliche Positionen in der BSEG ermittelt werden. Die Lösung dazu ist die Nutzung des SELECT-Befehls, um alle Sätze mit den passenden Kriterien wie Belegnummer, Buchungskreis und andere aus der BSEG zu ermitteln. Dieses Verfahren ist natürlich um einiges umständlicher als die Nutzung des INNER..JOIN, kommt aber am Schluß zu dem gleichen Ergebnis. Im Ganzen gesehen stellt die Programmierung jedoch höchstens 10% aller Probleme bei der Entwicklung der Arbeit; die Logik, das Testen und die Fehlersuche die restlichen 90%.





## 11.3 Testumgebung

Wie bei jedem fertig gestellten System stellt sich auch bei SAP die Frage: Wie kann ich sicher sein, daß mein Programm das Richtige ausweist?

Weiterführend ist die Zielformulierung, daß diese Prüfung auch auf dem Entwicklungssystem funktioniert. Es ist natürlich einfacher, einen Test auf dem QS-System durchzuführen, doch dafür wird wieder das KTW und die Auftragsverwaltung benötigt.

Aus diesem Grunde ist es sinnvoll, in einer Testumgebung das Programm zu testen. Es soll Abweichungen ausweisen, also quasi Fehlbuchungen. In der Regel werden, wenn überhaupt, nur fast schon zu erwartende Fehler gemacht: Durch das Umsetzen verschiedener Feldinhalte durch den „Ausblender“ werden zwar Abweichungsmeldungen produziert, jedoch kommen die Fälle, in denen manuell der Feldinhalt manipuliert wird, fast nie vor. Desweiteren werden bei diesem Testverfahren immer die gleichen Felder berührt. Der Report soll jedoch einen Beleg auf über 100 Felder testen, die abweichend sein können.

Als einzige Lösung bleibt, die Input-Daten nach dem Buchen manuell so zu verändern, daß genau die Fehler auftreten müssen, die getestet werden sollen. Dies kann aber nur über einen Umweg gemacht werden.

Belegpositionen, die einmal verbucht sind, müssen physisch in der Eingabedatei verändert werden. Dafür gibt es einen Report, der die UNIX-Eingabedatei auslesen und verändern kann. So ist dann auch ein Testen in der Entwicklungsumgebung möglich.

In der Regel ist eine Software nie beim ersten Test perfekt. So kann beispielsweise der Fall auftreten, daß das Programm in einen Endlos-LOOP läuft. SAP „handlet“ seine Online-Reports so, daß nach 300 CPU-Sekunden (einstellbar je Server) der automatische Abbruch erfolgt. Um sich diese Warterei und die Ressourcenbelegung zu ersparen, ist es möglich, über Administrationswerkzeuge laufende Threads\* abubrechen.

Ärgerlich wird es, wenn der Report wirklich so lange braucht, um das Datenvolumen zu bewältigen und trotzdem mit einem Time-Out gestoppt wird.

Aber auch dafür bietet SAP eine Möglichkeit: Reports lassen sich im Hintergrundmodus abspielen, indem ihnen eine unbegrenzte Laufzeit eingeräumt wird.

Im Kapitel 9.5.1 wurde das Thema Testumgebung – Direktes Buchen – behandelt.

Bei diesem Verfahren ging es darum, Online einen Beleg zu buchen, und daraus eine UNIX-Datei zu generieren.

Dabei trat jedoch ein unerwartetes Problem auf: Der Beleg wurde gebucht, die Testdatei generiert und der Report mit der generierten Datei als Eingabedatei gestartet. Als Ergebnis wurde ein nicht vorhandener Beleg ausgewiesen.

Damit war ein Problem aufgetreten, das nicht originär in SAP begründet ist.

Da der Report den Beleg als nicht vorhanden ausgewiesen hat, war klar, daß die Belegkopftexte nicht übereinstimmten. Diese Programmstelle war die einzige, die einen Beleg als fehlend ausweist. Die Frage war also, warum die Belegkopftexte ungleich waren, obwohl die Eingabedatei aus genau dieser Buchung entstanden war.

Bei dem manuellen Dynprofüllen wurde als Belegkopftext 0815 eingegeben. Mit diesem Belegkopftext ist der Beleg auch über die Transaktion FB03 im SAP R/3 – System wiederzufinden. In der Datei auf Betriebssystemebene stand jedoch zu diesem Beleg der Kopftext *D0154000014100000068IMYAL*. Dieser setzt sich zusammen aus dem Systemalias, der Mandantenummer, dem Buchungskreis, der Belegnummer und dem User-Account.

Diesen neuen Belegkopftext generiert der Buchungsreport selbständig. Dies hat auch seine Richtigkeit, da der normalerweise eine Datei generiert, die als Eingabedatei für den Report RFBIBL00 dient. Dieser

<http://www.gm.fh-koeln.de/~faeskorn/diplom/protoqerakis/11.html> (6 von 8) [21.01.03 15:24:23]



gesucht werden, d.h. sie sind auswertbar.  
Das bisher verwendete Grafikformat ist OTF.

- Vor R/3 Release 3.0 wurden ausgehende Dokumente immer über den Spool erzeugt und abgelegt.
- Ab R/3 Release 3.0 können ausgehende Dokumente im PDF-Format auch ohne den Weg über den Spool abgelegt werden.
- Ab R/3 Release 3.1G wurden ausgehende Dokumente sowohl im OTF- als auch im PDF-Format ohne den Spool abgelegt.
- Ab R/3 Release 4.0B gibt es für ausgehende Dokumente ausschließlich das PDF-Format. Es werden keine ausgehenden Dokumente im OTF-Format mehr erzeugt.

Ausgehende Dokumente werden über den Funktionsbaustein ARCHIV\_CREATE\_TABLE im PDF-Format abgelegt. Das Ablegen erfolgt damit nicht mehr über:

- CFBC-Kommando\*
- Spool

Die R/3-Anwendung erzeugt ein ausgehendes SAPscript-Dokument:

- direkt oder
- indirekt über die Nachrichtensteuerung

Von SAPscript wird ein Funktionsbaustein aufgerufen, der SAPscript-Daten übergeben bekommt. Diese Daten werden direkt abgelegt, so daß keine Datei mehr im Filesystem erzeugt wird.

Das externe Ablagesystem meldet das erfolgreiche Ablegen an SAP ArchiveLink zurück, d.h. die Ablage und die Dokument-ID. SAP ArchiveLink erzeugt daraufhin die Verknüpfung zwischen dem abgelegten Dokument und dem Business Objekt aus der R/3-Anwendungskomponente.

An dieser Stelle kommt das Migrationsproblem auf. Bei Migration auf Release-Stand 4.0 wird nur noch das PDF-Format unterstützt.

Die SAP AG läßt die Anwender jedoch im Unklaren darüber, was mit den Daten geschieht, die mit den 3.x-Release-Versionen archiviert wurden und im OTF-Format abgelegt sind.

Solange, wie diese Frage noch offen ist, und selbst von SAP-Beratern nicht definitiv beantwortet werden kann, ist das Einführen einer endgültigen Archivierungsroutine nicht realisierbar.

Dadurch stellt sich natürlich auch das nächste Problem: die Reorganisation der BI-Protokolldatei.

Wie bereits in der Einleitung dieser Arbeit angesprochen, ist das eigentliche Ziel, die mit dem Report erstellte Liste zu archivieren und die BI-Mappen und die zugehörigen BI-Protokollsätze zu löschen.

In Anbetracht der Unsicherheit, ob sich die Dokumente nach einem Migrationswechsel auf Anfrage der Revision überhaupt zurückladen lassen, ist es z.Zt. noch nicht verantwortbar, diese Reorganisation durchzuführen.

Der letzte Stand der Technik ist, daß ein Rückladen nicht möglich ist, eine Einsicht in diese archivierten Dokumente ist jedoch schon realisiert.

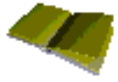
In Anbetracht der oben geschilderten Problematik konnten keine Produktivdaten gelöscht werden. Aus diesem Grunde wurde ein Testlauf mit Testdaten auf dem Entwicklungssystem durchgeführt, um die Funktionalität des Reorganisationsreports zu testen.

Ausgangspunkt waren drei BI-Mappen, die abgespielt wurden. Die Belege wurden sauber verbucht und die

Mappen bekamen den Status Verarbeitet.

Die soeben abgespielten Mappen wurden manuell gelöscht, die Protokolle blieben jedoch unangetastet. Der Reorganisationslauf wurde gestartet und die Protokolle wurden gelöscht. Anhand der Dateigröße der BI-Protokollsatzdatei wurde nachgewiesen, daß tatsächlich eine Reorganisation stattgefunden hatte.

Mit diesem Test wurde die Funktionalität sichergestellt. Die Zeit, die zur Reorganisation gebraucht wurde, hielt sich in sehr akzeptablem Rahmen, so daß ein solcher Lauf bedenkenlos jeden zweiten Tag nach dem Mappenlöschen gestartet werden kann.



# Schluß

## Zusammenfassung

## Ausblick

---

12.1 Zusammenfassung Die BAYER AG in Leverkusen setzt neben anderen Buchungssystemen seit Jahren SAP R/3 ein. Der Fachbereich IM-KSB betreut u.a. sowohl das Modul FI des SAP R/3-Systems als auch das Legacy-System MBS. Buchungen aus Fremdsystemen werden über eine Batch-Input-Schnittstelle ins SAP-System eingespielt. Dabei werden BI-Mappen erzeugt, die dann im Job von einem eigens für die Batchverarbeitung angelegten Batch-User abgespielt werden.

Jede Mappe, die im System erstellt und abgespielt wird, erzeugt Einträge in den APQ-Tabellen (BI-Mappen-Verwaltung) und in der BI-Protokolldatei. In der Protokolldatei stehen u.a. Warnungen und gelöschte Transaktionen. Aufgrund der Tatsache, daß täglich viele hundert Belege eingespielt werden, unterliegt diese Protokolldatei einem starken Wachstum. Hierbei handelt es sich um eine sequentielle Datei, die auf Betriebssystemebene steht. Sie ist weder manuell auswertbar, noch ist sie im Ganzen oder partiell archivierbar. Da das Protokoll nur lesbar ist, wenn es einen gültigen Verweis auf einen existierenden APQ-Eintrag enthält, müssen Protokoll und APQ-Tabellen immer gemeinsam gehalten werden. Die Größe dieser beiden Objekte umfaßt bereits viele GByte, was das System langsam und auf Dauer kaum noch reorganisierbar macht.

Die einfachste Lösung für dieses Ressourcenproblem ist es, nach dem Abspielen Mappen und Protokolle zu löschen.

Dies ist jedoch nicht zulässig, da während des Abspielens der BI-Mappen Feldinhalte oder Transaktionen geändert oder gelöscht werden können. Anhand des wenig aussagekräftigen Protokolls würden einige dieser Änderungen ausgewiesen. Dieser Nachweis ist mehrere Jahre aufbewahrungspflichtig; daher müssen sämtliche Einträge im System gehalten bzw. individuell archiviert werden. Aufgrund dieser Situation ergab sich die Anforderung, eine inhaltlich befriedigende Lösung mit SAP-Mitteln zu entwickeln.

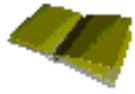
Das Verbuchen der BI-Daten mußte auf andere Weise protokolliert werden, um die Einträge in den Tabellen und im Protokoll zu löschen.

Diese Aufgabe wurde gelöst, indem im Rahmen der Diplomarbeit ein Report auf Basis der Programmiersprache ABAP/4 entwickelt wurde.

Diese Eigenentwicklung vergleicht die täglichen MBS-Buchungen, die in der BI-Schnittstellendatei gemäß SAP-Strukturaufbau angeliefert werden, mit den tatsächlich im SAP-System verbuchten Belegen. Dabei vergleicht er die Buchungen auf Feldebene pro Datensatz und prüft, ob die Feldinhalte in BI-

Schnittstellendatei und SAP-System übereinstimmen.

Das Protokoll weist Abweichungen, korrekt verbuchte und nicht verbuchte Belege separat aus.



## 12.2 Ausblick

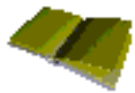
Bei der im Rahmen der Diplomarbeit erstellten Lösung wurde ein Konzept entwickelt, das auf eine bestimmte Problemstellungen hin entstanden ist.

Die Idee und der Lösungsansatz ist jedoch durchaus auf analoge Probleme aus anderen Arbeitsfeldern übertragbar, da es um das Umfeld des Batch-Input geht. Batch-Input ist ein Verfahren, das nicht nur bei der vorliegenden Aufgabe, sondern auch beim Einspielen von Daten aus anderen Quellen angewendet wird. Es wird u.U. nur eine abgewandelte Technik angewendet werden müssen, die dem Systemumfeld optimaler angepaßt ist.

Um das Zusammenspiel zwischen dem Einspielen der Fremddaten ins System über Batch-Input, dem erstellten Abstimmreport und dem Reorganisieren der BI-Protokolldatei sicherzustellen, ist es notwendig, die einzelnen Arbeitsschritte in einen Batch-Lauf zu integrieren. Nach dem Buchungsschluß in den Fremdsystemen werden automatisch Batch-Input-Mappen generiert, die im Hintergrund abgespielt werden. Sobald der Buchungslauf erfolgt ist, wird der Abstimmreport gestartet und damit die Liste erzeugt. Anschließend werden die Protokolle der verbuchten Mappen gelöscht und der Standardreport zur Reorganisation gestartet.

Das komplette Verfahren soll als Batch-Lauf organisiert werden, so daß am nächsten Morgen alle Ergebnisse vorliegen.

Entsprechende Versuche sind für den Herbst geplant, wenn die Umstellung auf Release 4.0 und der Y2K-Test erfolgreich absolviert ist.



# Anhang

## Begriffserklärungen

## Literaturverzeichnis

## Abbildungsverzeichnis

## Tabellenverzeichnis

## Struktogrammverzeichnis

### 13 Anhang

#### 13.1 Begriffserklärungen

ABAP/4	Advanced Business Application Programming - Programmiersprache der vierten Generation, die SAP zur Entwicklung von Anwendungsprogrammen entwickelt hat.
ABAP/4-Dictionary	Das ABAP/4-Dictionary gewährleistet Redundanzfreiheit der Datenbanken, die Datenintegrität* und die Datenkonsistenz, indem es jede Information nur einmal erfaßt und die Informationen an jeder Stelle im System zur Verfügung stellt.
ALE	Application Link Enabling - bietet die Grundlagen für die Integration von Geschäftsprozessen, die über mehrere R/3- oder Fremdsysteme hinweg abgewickelt werden können. Dabei werden Daten zwischen Anwendungssystemen kontrolliert ausgetauscht und konsistent gehalten. ALE unterstützt Konfiguration und Betrieb von verteilten Anwendungen. Es umfaßt einen betriebswirtschaftlich kontrollierten Nachrichtenaustausch zwischen verteilten Anwendungen bei konsistenter Datenhaltung auf lose gekoppelten R/3-Systemen.  (siehe Exkurs)
BAPI	Business Application Origramming Interface – über diese Schnittstelle kann von außen auf Geschäftsprozesse und Business-Objekte zugegriffen und so eine direkte Kommunikation zwischen Anwendungen unterschiedlicher Softare-Anbieter ermöglicht werden.
Baustein FI	Ist das Buchhaltungsmodul im SAP R/3-System

Bewegungsdaten	<p>Vorgangsbezogene Daten, die kurzlebig sind und bestimmten Stammdaten zugeordnet werden.</p> <p>Einzelne Buchungsbelege werden als Bewegungsdaten bezeichnet. Beispiele dafür können Lagerbestandsbuchungen sein.</p>
BI-Mappe	<p>Zusammenfassung einer Folge von Transaktionen, die durch ein Programm mit Anwenderdaten versehen wurden. Diese Transaktionen werden als Stapel gespeichert und die Mappe kann zu einem späteren Zeitpunkt im Online-System abgespielt werden.</p> <p>Erst beim Abspielen der Mappe finden Datenbankveränderungen statt.</p> <p>Durch dieses Verfahren können in kurzer Zeit große Datenmengen in das SAP-System eingespielt werden.</p>
CFBC	Queue-Verwaltung: Fehlerbehandlung
Customizing	Methode im R/3-System, mit der man die unternehmensneutral ausgelieferte Funktionalität den spezifischen betriebswirtschaftlichen Anforderungen des Unternehmens anpaßt.
Dateninkonsistenz	Daten sind nicht mehr in sich schlüssig oder nur noch unvollständig bzw. ohne Zusammenhang.
Datenintegrität	Die Korrektheit und die Konsistenz der Daten in einer Datenbank.
Datenredundanz	Mehrfaches Speichern desselben Datums an verschiedenen Orten.
DDIC	<p><b>Data Dictionary</b> - SAP kann als Datenbank ausschließlich relationale Datenbanksysteme verwenden, z.B. Oracle oder Informix. Das Data-Dictionary ist eine zentrale Informationsquelle über die Daten eines DV-Systems.</p> <p>Es dient zunächst der Erfassung und Verwaltung von Datendefinitionen, sog. Metadaten. Außerdem stellt ein DDIC die Informationsbasis für Benutzer sowie für verschiedene Softwarekomponenten in einem modernen Datenverwaltungssystem dar. Informationen, wie z.B. welche Daten in welchen Datenbanken eines Unternehmens enthalten sind, welche technischen Eigenschaften die Daten haben, u.v.a. können sehr schnell abgerufen werden.</p>
Debitor	Geschäftspartner, gegenüber dem Forderungen für erbrachte Leistungen bestehen (Kunde).

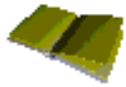
Dispatcher	Jeder Applikationsserver enthält einen Dispatcher. Der Dispatcher ist das Bindeglied zwischen den Workprozessen und den an dem Applikationsserver angemeldeten Benutzern (bzw. deren SAP GUIs). Seine Aufgabe besteht darin, jedesmal, wenn eine Benutzerinteraktion dazu führt, daß ein Dialogschritt ausgeführt werden soll, diesen Auftrag vom SAP GUI entgegenzunehmen und ihn an einen freien Workprozeß weiterzuleiten. Auf dem gleichen Wege gelangen die Bildschirmausgaben, die aus der Ausführung des Dialogschritts resultieren, wieder zum Benutzer zurück.
Dynpro	<p>DYNamisches PROgramm, das aus einem Bildschirmbild und der unterliegenden Ablauflogik besteht. Die Hauptkomponenten eines Dynpros sind:</p> <p>Attribute (z.B. Dynpronummer, Folgedynpro)</p> <p>Layout (Anordnung der Texte, Felder und der anderen Elemente)</p> <p>Feldattribute (Definition der Eigenschaften der einzelnen Felder)</p> <p>Ablauflogik (Aufrufe der relevanten ABAP-Module)</p>
EPK	<b>E</b> reignisgesteuerte <b>P</b> rozess <b>K</b> ette – Anhand von Ereignissen und Ergebnissen werden Geschäftsprozesszusammenhänge mit Hilfe von Informationsflüssen dargestellt.
ERP-Standardsoftware	<b>E</b> nterprise <b>R</b> esource <b>P</b> laning – internationaler Begriff für betriebswirtschaftliche Standardsoftware
Geschäftsprozesse	Ein Geschäftsprozess ist eine zeitlich-logische Folge von Prozessen und Funktionen, die auf eine externe oder interne Zielgruppe zur Erbringung einer definierten Leistung ausgerichtet sind. Im R/3-Referenzmodell werden Mustergeschäftsprozesse in Form von Szenarien als EPKs zur Verfügung gestellt.
HGB	HandelsGesetzBuch
Host	Großrechner (IBM, MVS-Betriebssystem)
IDOC	<p>Intermediate Document – Datencontainer für den Austausch zwischen SAP-Systemen bzw. zwischen SAP-Systemen und Fremdsystemen. Die Idocs besitzen eine neutrale Datenstruktur.</p> <p>Fremdsysteme können Idocs als Standard-Schnittstelle zur Datenübergabe nutzen.</p>

Interne Tabelle	<p>Datenstruktur, die nur während der Laufzeit eines Programms existiert. Die interne Tabelle ist einer von zwei strukturierten Datentypen in ABAP. Sie besteht aus einer beliebigen Anzahl identisch aufgebauter Tabellenzeilen mit oder ohne eine Kopfzeile.</p> <p>Die Kopfzeile entspricht einer Feldleiste und dient als Arbeitsbereich für die interne Tabelle. Der Datentyp der Zeilen kann elementar oder strukturiert sein.</p>
Kreditor	Geschäftspartner, gegenüber dem Verbindlichkeiten für erhaltene Leistungen bestehen (Lieferant).
Legacy	Altsysteme auf dem Host
MBS	<b>M</b> odulares <b>B</b> uchungs <b>S</b> ystem, das auf einem Großrechner in einer HOST-Umgebung läuft (Eigenentwicklung COBOL/IMS).
PAI	<b>P</b> rocess <b>A</b> fter <b>I</b> ntput – Verarbeitungsblock, der nach der Anzeige eines Dynpros verarbeitet wird.
PBO	<b>P</b> rocess <b>B</b> efore <b>O</b> utput – Verarbeitungsblock, der nach dem Aufruf eines Dynpros aber vor der eigentlichen Bildanzeige verarbeitet wird.
Redundanz	Doppeltes Auftreten eines Datum
Repository	<p>Ein System, das Informationen über Objekte der Softwareproduktion und deren Beschreibung und Beziehungen untereinander verwaltet, auswertet und bereitstellt.</p> <p>Umfaßt das Data Dictionary und die Entwicklungsumgebung, d.h. sämtliche Informationen über ein Objekt, die während eines Entwicklungsprozesses entstehen.</p>
Sachkonto	<p>Struktur zur Aufnahme von Wertebewegungen in einem Buchungskreis bezüglich einer Sachkontenposition eines Kontenplans.</p> <p>Sie umfaßt Verkehrszahlen, in denen Wertebewegungen einer Buchungsperiode für Berichtszwecke des Hauptbuchs, die ein Sachkonto und ggf. einen Geschäftsbereich betreffen, als Summenwerte dargestellt sind.</p>
SAM	<b>S</b> equential <b>A</b> ccess <b>M</b> emory
SAPGUI	SAP-eigenes Graphical User Interface
SAP-SAP	Datenübertragung von einem Arbeitsfeld des R/3 mit Hilfe von ALE in ein anderes.



Sekundärindex	<p>Im ABAP Dictionary definierter Index zu einer Tabelle.</p> <p>Beim Aktivieren einer Tabelle im ABAP Dictionary wird automatisch ein Index über die Primärschlüsselfelder (Primärindex) der Tabelle angelegt.</p> <p>Falls weitere Indizes zur Tabelle benötigt werden, müssen diese als Sekundärindizes angelegt werden.</p>
Selektionsparameter	Auf dem Selektionsbild kann man die Selektionsparameter eingeben, nach denen der Report eine Auswertung der Datenbank durchführt und eine Ergebnisliste erzeugt.
Sequentiell	Satz für Satz ohne direkten Zugriff
Sonderhauptbuchvorgang	<p>Spezieller Vorgang in der Debitoren- und Kreditorenbuchhaltung, der im Haupt- und Nebenbuch gesondert ausgewiesen wird.</p> <p>Dazu zählen u.a. Wechsel, Anzahlungen, Bürgschaften.</p>
Stammdaten	Informationen, die dauerhaft hinterlegt sind, und sich fast nie ändern, z.B.: Geburtsdatum, Konfession
Thread	Prozesse, die für sich eigenständig ablaufen.
Transaktion	<p>Logisch abgeschlossener Vorgang (Einheit) im R/3-System; aus Sicht der Dialogprogrammierung ist sie ein komplexes Objekt, das aus einem Modulpool und Dynpros besteht und mit einem vierstelligen Transaktionscode aufgerufen wird.</p> <p><u>Ablauf einer Transaktion:</u></p> <p>Das Dialogprogramm stellt die Bildschirmmasken zur Verfügung</p> <p>Der Anwender gibt Daten ein und beendet den Dialog durch die Return-Taste oder den OK-Button</p> <p>Die Eingaben werden vom Dialogprogramm überprüft. Sind sie korrekt, wird ein Protokollsatz erzeugt und in die Protokollsatzdatei abgespeichert</p> <p>Das Verbuchungsprogramm liest die Protokollsatzdatei und beauftragt den Datenbankserver mit den entsprechenden Änderungen der Datenbank</p> <p>Nach erfolgreicher Änderung wird der Protokollsatz aus der Protokollsatzdatei gelöscht</p>

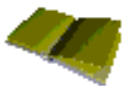
Workflow	<p>Ausführung einer Mehrschrittaufgabe zur Laufzeit.</p> <p>Ein Workflow besteht aus einer Abfolge von Schritten, die entweder von Personen oder automatisiert vom System bearbeitet werden.</p> <p>Die zeitliche und logische, an die Auswertung von Bedingungen geknüpfte Reihenfolge der Schritte wird vom Workflow-Manager kontrolliert und kann durch ereignisbezogene Reaktionsmechanismen flexibel gesteuert werden.</p> <p>Technisch gesehen wird eine Workflow-Instanz durch ein Workflow-Item (Workitem vom Typ F) repräsentiert.</p>
Workprozeß	<p>Applikationsserver enthalten Workprozesse, deren Anzahl und Typ beim Start des R/3-Systems festgelegt wird. Workprozesse sind Komponenten, die in der Lage sind, eine Anwendung (bzw. jeweils einen Dialogschritt) auszuführen. Jeder Workprozeß hat Verbindung zu einem Speicherbereich in dem sich der Kontext der gerade ausgeführten Anwendung befindet. Der Kontext beinhaltet die aktuellen Daten des laufenden Programms und muß für jeden Dialogschritt dieses Programms verfügbar sein</p>



## 13.2 Literaturverzeichnis

[1] SAP R/3 kompakt	Einführung und Arbeitsbuch für die Praxis
Engels, Gensch, Nottenkämper	tewi Verlag GmbH, 1996
[2] SAP R/3 Basissystem	Praxistrainer
Herth	Markt & Technik, 1996
[3] Praxistrainer SAP R/3	Grundlagen, Architektur, Anwendung

	v. Vorkampff-Laue	Markt & Technik, 1994
[4] SAP R/3 Finanzwesen		Grundlagen, Anwendung, Fallbeispiele
	Hefner	Markt & Technik, 1996
[5] BAYER-Intranet		Stand: März 1999
[6] Vorlesungsskript Victor		Wintersemester 98/99
[7] Archivieren in SAP R/3		Dokumentation 3574 der BAYER AG
	B. Kubner	21. April 1999
[8] SAP R/3, Technischer Leitfaden Datenarchivierung		Dokumentation 3575 der BAYER AG
	B. Kubner	21. April 1999
[9] SAP Doku Print 3.1g		SAP R/3 – System
		Mai 1999
[10] SAP R/3		Kompendium
	R. Möhrten, F. Kokot	Markt & Technik 1998
[11] SAPNET		



## 13.3 Abbildungsverzeichnis

Abbildung 1 - Allgemeine Bayer-Konzern-Übersicht [5]

Abbildung 2 - Arbeitsfelder der BAYER AG [5]

Abbildung 3 – Aufbau und Funktionalität von SAP R/3 [5]

Abbildung 4 - Die 3 Ebenen (Server) des R/3-Systems [6]

Abbildung 5 - Dispatching von Dialogschritten [5]

Abbildung 6 - Die vier Typen des C/S [6]

Abbildung 7 - Zusammenspiel Belegkopf mit Position

Abbildung 8 - Organisation FI - Grundstruktur [10]

Abbildung 9 – Ablauf des Batch-Input-Verfahrens

Abbildung 10 - Parameterauswahl des RSBDCREO

Abbildung 11 - Update-Service [2]

Abbildung 12 - Schematischer Ablauf eines BI-Verfahrens

Abbildung 13 – Startbild RFBIBL00

Abbildung 14 - Schematischer Ablauf der Anwendungen und Objekte

Abbildung 15 - Mappe mit manuellen Eingriff

Abbildung 16 - Zusammenspiel ADK und ArchivLink [7]

Abbildung 17 – Laufzeiten [9]

Abbildung 18 - Inhalt einer BI-Schnittstellendatei

Abbildung 19 - Freigegebener Auftrag

Abbildung 20 - ABAP/4-Programmtypen [2]

Abbildung 21 - Statuszeile FB01 nach erfolgreicher Buchung

Abbildung 22 - ALE–Grobstruktur [9]

Abbildung 23 - Eingabemaske des Reports

Abbildung 24 - Ergebnisbild einer DD03L-Anfrage

Abbildung 25 - Ausgewiesene Belegabweichungen

Abbildung 26 - Inhalt eines BBKPF nach dem Ausblenden

Abbildung 27 - Ausnahmenbekanntmachung zu Beginn einer Liste

Abbildung 28 - Listenanzeige der nicht vorhandenen Belege

Abbildung 29 - Listanzeige der korrekten Belege

Abbildung 30 - Statistik der gelesenen Belegsätze

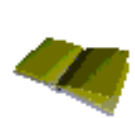
Abbildung 31 - Felddefinition eines Betragsfeldes BBSEG und BSEG

Abbildung 32 - Fehlerhaftes Ausweisen eines Feldes in der Report-Liste

Abbildung 33 - Feldvergleich zur Laufzeit des Reports per Debugger

Abbildung 34 – Eigenschaft der Domäne KOSTL

Abbildung 35 - Inhalt der BI-Protokolldatei



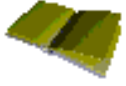
## 13.4 Tabellenverzeichnis

Tabelle 1 - Segmente des Objektes FI\_DOCUMNT

Tabelle 2 - Merkmale für Archivierungsobjekte

Tabelle 3 - Beispiele für MBS-Daten verschiedener Landeswährungen

Tabelle 4 - Feldaufbauvergleich zweier gleichnamiger Felder



## 13.5 Struktogrammverzeichnis

Struktogramm 1 - Struktureller Ablauf des Reports

Struktogramm 2 - Ablauf der Feldvergleiche MBS-SAP

